WSN INTEGRATION WITH CLOUD COMPUTING TO CREATE MONITORING SYSTEM FOR WATER & UNDERWATER ENVIRONMENT

PROJECT REPORT

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

Atishay Arvind Singh(121274)

Under the Supervision of

Mr. Ravindara Bhatt

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"WSN INTEGRATION WITH CLOUD COMPUTING TO CREATE MONITORING SYSTEM FOR WATER & UNDERWATER ENVIRONMENT"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2016 to June 2016 under the supervision of **Mr. Ravindara Bhatt** Assistant Professor (Senior Grade).
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Atishay Arvind Singh [121274]

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Ravindara Bhatt
Assistant Proffesor (Senior Grade)
Computer Science and Engineering
Dated:

ACKNOWLEDGEMENT

This project work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost I offer my sincerest gratitude to my supervisor **Mr. Ravindara Bhatt** who has supported me throughout my project work with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the completion of my final year project to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

The work was performed at Department of Computer Science and Engineering at the University, and I would like to thank all the people there for their hospitality and support.

Date: ______________ ________________

ABSTRACT

Three-Dimensional Wireless Sensor Networks (3-D WSNs) have the ability to deal with multimedia content such as audio, video, still images and scalar sensor data. Three-dimensional WSNs based surveillance systems can detect occurrences of events at designated points over time. Some of the leading applications include surveillance applications, battle-space monitoring, human tracking, underwater monitoring and urban warfare. The system presents an integrated framework of three-dimensional wireless sensor network (3-D WSN) with cloud computing for underwater surveillance applications. The integrated framework provides a faster and more convenient platform for the client to obtain information from an array of sensor nodes that has been set-up for monitoring and surveillance marine environment. At the server side, the sensor nodes collect the data and transfer them to cloud where the data are stored in the databases and then processed. On the client side, the user will use the internet to request for the web services that will fetch the data from the distributed SQL databases and on the basis of the threshold values would display the warnings as well as the data obtained. In addition, this work presents the concept of cloud computing and services. The benefits of this system include basic computing hardware and reasonable storage capacities making it suitable for any 3D WSN which provides real time monitoring and surveillance in 3-D space. The customers can fully access our cloud service using devices that have internet capabilities. This work focuses includes distributed computing and resource allocation of processing three-dimensional wireless multimedia sensor data, collected in a 3-d space, and performance evaluation over various parameters settings in a cloud environment. We utilize CloudSim for simulating the cloud computing environment.

LIST OF FIGURES

Fig 1: Cloud Architecture 6

Fig 2: Communication architecture using a gateway 10

Fig 3: Communication Architecture with direct Connections 10

Fig 4: Sensor Network backbone Infrastructure 11

Fig 5: Underwater Deployment Strategies 12

Fig 6: Node Deployment Strategies 12

Fig 7: Layered CloudSim architecture 14

Fig 8: Proposed System Architecture 16

Fig 9: Common Architecture of 3D monitoring system 17

Fig 10: Common Architecture of UWSN Node 17

Fig 11: Underwater Scenario Model 19

Fig 12: Cloud System Structure 23

Fig 13: Sequence Diagram 24

Fig 14: Host to VM Mapping 25

Fig 15: Class Design Diagram of CloudSim 26

Fig 16: VM State Transition Diagram 27

Fig 17: Effects of Different Provisioning Polices on task unit execution 27

Fig 18: Throughput with 2 Cores 29

Fig 19: Throughput in Time Shared VMAllocation Policy 30

Fig 20: Avg Waiting Time with 2 Cores 30

Fig 21: Avg Waiting Time in Time Shared VMAllocation Policy 31

Fig 22: Effect of varying Ram 31

Fig 23: Received Packet Breakdown 32

Fig 24: Packet Received per Node 32

Fig 25: Application Level Latency 33

Fig 26: Received Packet Breakdown 33

Fig 27: Packet Received per Node 34

Fig 28: Application Level Latency 34

Fig 29: Received Packet Breakdown 35

Fig 30: Packet Received per Node 35

Fig 31: Application Level Latency 36

Fig 32: % Packets Delivered 37

LIST OF TABLES

Table 1 Various Communication Techniques 18

Table 2 Various Sensors Available 19

Table 3 Cloud Simulation Environment 29

TABLE OF CONTENTS

Certificate

Acknowledgement

Abstract

List of Figures

List of Tables

CHAPTER-1

INTRODUCTION **1-9**

APPLICATIONS OF WIRELESS SENSOR NETWORK 1

TYPES OF WSNS (WIRELESS SENSOR NETWORKS) 2

UNDERWATER WIRELESS SENSOR NETWORKS 4

CLOUD COMPUTING 5

PROBLEM STATEMENT 7

OBJECTIVES 8

ORGANIZATION 9

CHAPTER-2

LITERATURE SURVEY **10-15**

LITERATURE REVIEW 10

CHAPTER-3

PROPOSED FRMEWORK **16-28**

GENERAL SENSOR NODE ARCHITECTURE 16

WIRELESS COMMUNICATION TECHNOLOGIES 18

SENSING PARAMETERS AND SENSORS 19

UNDERWATER WIRELESS SENSOR NETWORK ARCHITECTURE 19

ROUTING PROTOCOLS FOR UWSN 21

CLOUD SERVER ARCHITECTURE 22

DATACENTER AND VM ARCHITECTURE 26

CHAPTER-4

PERFORMANCE ANALYSIS **29-37**
CLOUD SIMULATION 29
UWSN SIMULATION 31

CHAPTER-5
CONCLUSION **38**

REFERENCES **39**

INTRODUCTION

Sensing is a technique used to collect information about a physical object or process, including the happening of events such as changes in state like dropping in temperature or pressure. An object performing a sensing task is called a sensor. For example, the human body is supplied with sensors that can capture optical information from the environment (eyes), acoustic information such as sounds (ears), and smells (nose). These are examples of remote sensors, i.e. they do not require touching the monitored object to gather information. From a technical view, a sensor is a device that translates events or parameters in the real world into signals that can be determined and analyzed.
Wireless sensor networks (WSN), sometimes called wireless sensor and actuator networks (WSAN) are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

APPLICATIONS OF WIRELESS SENSOR NETWORK

Area Monitoring
Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors detect enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines.

Health care Monitoring
The medical applications can be of two types: wearable and implanted. Wearable devices are used on the body surface of a human or just at close proximity of the user. The implantable medical devices are those that are inserted inside human body. There are many other applications too e.g. body position measurement and location of the person, overall monitoring of ill patients in hospitals and at homes. Body-area networks can collect information about an individual's health, fitness, and energy expenditure.

Environmental/Earth sensing

There are many applications in monitoring environmental parameters, examples of which are given below. They share the extra challenges of harsh environments and reduced power supply.

Air pollution monitoring

Wireless sensor networks have been deployed in several cities (Stockholm, London, and Brisbane) to monitor the concentration of dangerous gases for citizens. These can take advantage of the ad hoc wireless links rather than wired installations, which also make them more mobile for testing readings in different areas.

Forest fire detection

A network of Sensor Nodes can be installed in a forest to detect when a fire has started. The nodes can be equipped with sensors to measure temperature, humidity and gases which are produced by fire in the trees or vegetation. The early detection is crucial for a successful action of the firefighters; thanks to Wireless Sensor Networks, the fire brigade will be able to know when a fire is started and how it is spreading.

Water quality monitoring

Water quality monitoring involves analyzing water properties in dams, rivers, lakes & oceans, as well as underground water reserves. The use of many wireless distributed sensors enables the creation of a more accurate map of the water status, and allows the permanent deployment of monitoring stations in locations of difficult access, without the need of manual data retrieval. [8]

Industrial monitoring

Machine health Monitoring

Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionality. Wireless sensors can be placed in locations difficult or impossible to reach with a wired system, such as rotating machinery and untethered vehicles.

Water/Waste water Monitoring

Monitoring the quality and level of water includes many activities such as checking the quality of underground or surface water and ensuring a country's water infrastructure for the benefit of both human and animal. It may be used to protect the wastage of water.

Structural health Monitoring

Main article: Structural health monitoring

Wireless sensor networks can be used to monitor the condition of civil infrastructure and related geo-physical processes close to real time, and over long periods through data logging, using appropriately interfaced sensors.

TYPES OF WSNS (WIRELESS SENSOR NETWORKS)

Depending on the environment, the types of networks are decided so that those can be deployed underwater, underground, on land, and so on. Different types of WSNs include:

- Terrestrial WSNs
- Underground WSNs
- Underwater WSNs
- Multimedia WSNs
- Mobile WSNs

1. TERRESTRIAL WSNs

Terrestrial WSNs are capable of communicating base stations efficiently, and consist of hundreds to thousands of wireless sensor nodes deployed either in unstructured (ad hoc) or structured (Preplanned) manner. In an unstructured mode, the sensor nodes are randomly distributed within the target area that is dropped from a fixed plane. The preplanned or structured mode considers optimal placement, grid placement, and 2D, 3D placement models.

In this WSN, the battery power is limited; however, the battery is equipped with solar cells as a secondary power source. The Energy conservation of these WSNs is achieved by using low duty cycle operations, minimizing delays, and optimal routing, and so on.

2. UNDERGROUND WSNs

The underground wireless sensor networks are more expensive than the terrestrial WSNs in terms of deployment, maintenance, and equipment cost considerations and careful planning. The WSNs networks consist of a number of sensor nodes that are hidden in the ground to monitor underground conditions. To relay information from the sensor nodes to the base station, additional sink nodes are located above the ground. The underground wireless sensor networks deployed into the ground are difficult to recharge. The sensor battery nodes equipped with a limited battery power are difficult to recharge. In addition to this, the underground environment makes wireless communication a challenge due to high level of attenuation and signal loss.

3. UNDERWATER WSNs

More than 70% of the earth is occupied with water. These networks consist of a number of sensor nodes and vehicles deployed under water. Autonomous underwater vehicles are used for gathering data from these sensor nodes. A challenge of underwater communication is a long propagation delay, and bandwidth and sensor failures. Under water WSNs are equipped with a limited battery that cannot be recharged or replaced. The issue of energy conservation for under water WSNs involves the development of underwater communication and networking techniques.

4. MULTIMEDIA WSNs

Multimedia wireless sensor networks have been proposed to enable tracking and monitoring of events in the form of multimedia, such as imaging, video, and audio. These networks consist of low-cost sensor nodes equipped with microphones and cameras. These nodes are interconnected with each other over a wireless connection for data compression, data retrieval and correlation.

The challenges with the multimedia WSN include high energy consumption, high bandwidth requirements, data processing and compressing techniques. In addition to this, multimedia contents require high bandwidth for the contents to be delivered properly and easily.

5. MOBILE WSNs

These networks consist of a collection of sensor nodes that can be moved on their own and can be interacted with the physical environment. The mobile nodes have the ability to compute sense and communicate. The mobile wireless sensor networks are much more versatile than the static sensor networks. The advantages of MWSN over the static wireless sensor networks include better and improved coverage, better energy efficiency, superior channel capacity, and so on.

UNDERWATER WIRELESS SENSOR NETWORKS

Underwater Acoustic Sensor Networks (UW-ASN) consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network which can adapt to the characteristics of the ocean environment. Underwater networking is a rather unexplored area although underwater communications have been experimented since World War II, when, in 1945, an underwater telephone was developed in the United States to communicate with submarines. Acoustic communications are the typical physical layer technology in underwater networks. In fact, radio waves propagate at long distances through conductive sea water only at extra low frequencies (30-300 Hz), which require large antennae and high transmission power. Optical waves do not suffer from such high attenuation but are affected by scattering. Moreover, transmission of optical signals requires high precision in pointing the narrow laser beams. Thus, links in underwater networks are based on acoustic wireless communications.

The traditional approach for ocean-bottom or ocean column monitoring is to deploy underwater sensors that record data during the monitoring mission, and then recover the instruments. This approach has the following disadvantages:

- Real time monitoring is not possible. This is critical especially in surveillance or in environmental monitoring applications such as seismic monitoring. The recorded data cannot be accessed until the instruments are recovered, which may happen several months after the beginning of the monitoring mission.
- No interaction is possible between onshore control systems and the monitoring instruments. This impedes any adaptive tuning of the instruments, nor is it possible to reconfigure the system after particular events occur.
- If failures or misconfigurations occur, it may not be possible to detect them before the instruments are recovered. This can easily lead to the complete failure of a monitoring mission.
- The amount of data that can be recorded during the monitoring mission by every sensor is limited by the capacity of the onboard storage devices (memories, hard disks, etc.).

Therefore, there is a need to deploy underwater networks that will enable real time monitoring of selected ocean areas, remote configuration and interaction with onshore human operators. This can be obtained by connecting underwater instruments by means of wireless links based on acoustic communication.

Major challenges in the design of underwater acoustic networks are:

- Battery power is limited and usually batteries cannot be recharged, also because solar energy cannot be exploited;
- The available bandwidth is severely limited;
- Channel characteristics, including long and variable propagation delays, multi-path and fading problems;
- High bit error rates;
- Underwater sensors are prone to failures because of fouling, corrosion, etc.

Applications of underwater networks fall into similar categories as for terrestrial sensor networks. Scientific applications observe the environment: from geological processes on the ocean floor, to water characteristics (temperature, salinity, oxygen levels, bacterial and other pollutant content, dissolved matter, etc.) to counting or imaging animal life (micro-organisms, fish or mammals). Industrial applications monitor and control commercial activities, such as underwater equipment related to oil or mineral extraction, underwater pipelines or commercial fisheries. Industrial applications often involve control and actuation components as well. Military and homeland security applications involve securing or monitoring port facilities or ships in foreign harbors, de-mining and communication with submarines and divers.

With the increase in demand and hardware limitations and pursuit for better performance, we use the concept of cloud computing to achieve better performance and enhancing greater computing capability. The virtualization techniques in cloud computing is used so as to reduce the limitations posed by the hardware components. The increase in data manipulations from sensor networks into information requiring applications when integrated with the techniques of cloud computing would definitely benefit us.

CLOUD COMPUTING

Cloud computing also referred as the cloud (due to the internet's representation in flow diagrams) is an on-demand computing model which consists of independent, networked hardware and/or software resources. Cloud computing applies the concept of virtualization for optimal usage of hardware and/or software resources. In simple terms cloud computing is the virtualization of a pool of resources, under data centers for hosting cloud applications, which

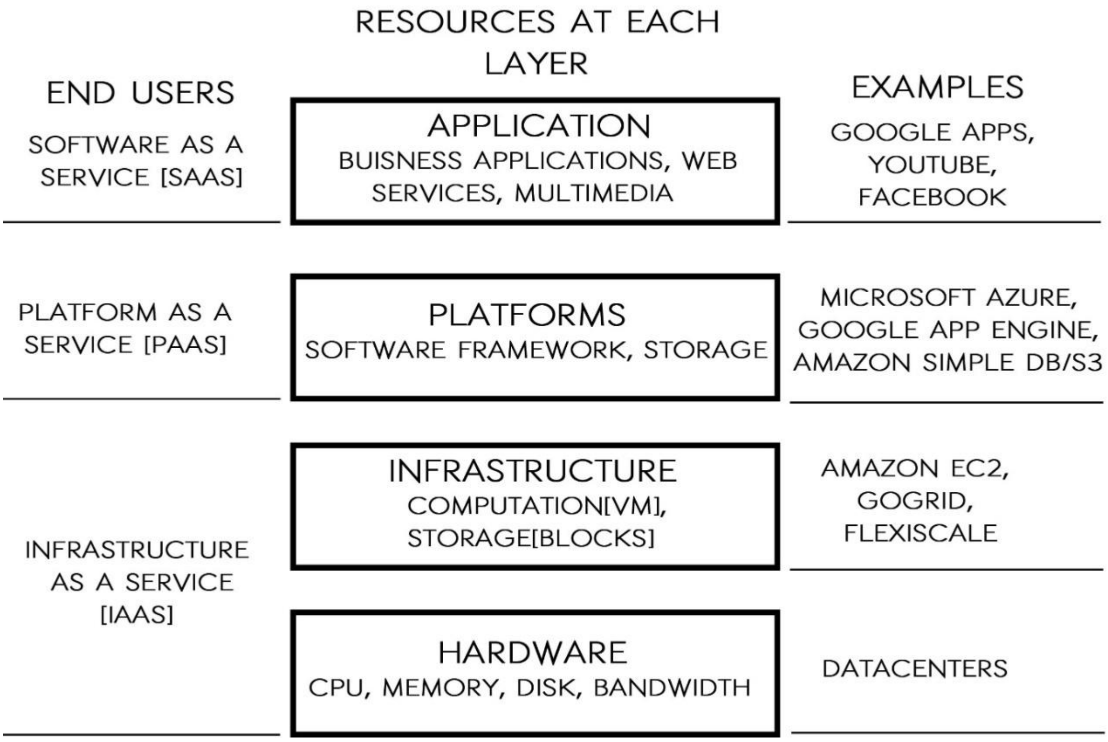

Fig 1: Cloud Architecture

Are made available to everyone on subscription basis. In cloud computing the hardware and software resources are made available by the providers for different users or clients. Service providers offer cloud services with predefined quality of service (QOS) terms through the Internet as a collection of easy-to use, scalable, and economically feasible services to the clients. The cloud services fall under three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The characteristics of cloud computing have attracted many IT giants like Amazon, Google, Microsoft, SharePoint, VMware etc. Amazon is the leader amongst all cloud providers. The two most common services they provide are Amazon S3 a Simple Storage Service and Amazon EC2 Elastic Cloud Computing, both belong to IaaS service model.S3 and EC2 both work on a concept of pay-as-you-go model. Therefore, the number of people using these services are exponentially increasing with the increase in deployment of new applications on the cloud as well.

PROBLEM STATEMENT:

To monitor underwater environment for impurities and undesirable substances so that we may conserve the potability of the water sources. This platform must be able to monitor various river cleaning schemes which are being employed.

The proposed solution integrates wireless sensor technology with cloud computing to provide us with an easy to use and access system for monitoring the underwater systems. Furthermore the system provides us with ready to deploy architecture for underwater WSN.

OBJECTIVES:

Main objectives of this project are:

- To create a method for deploying nodes in underwater scenario with maximum coverage.
- To create an architecture with routing and mac protocol for UWSN.
- To create an architecture for cloud computing system for processing the collected data at the nodes.

ORGANIZATION:

1. INTRODUCTION: It gives an overview and basic knowledge about the project. It contains an introduction to basics about WSN and cloud computing.

2. LITERATURE SURVEY: It includes facts and description obtained from various well published and authenticated research papers.

3. PROPOSED FRAMEWORK: It includes the System Analysis, UML diagrams and the algorithms implemented.

4. PERFORMANCE ANALYSIS: It includes comparison between FCFS and DVFS and also contains the outputs obtained on implementation of the virtual cloud.

5. CONCLUSION: The proposition of a system for water monitoring by using underwater wireless sensor networks is presented.

LITRATURE SURVEY

I] Advances in hardware technology and wireless communications will enable the development of large-scale wireless sensor networks (WSN). Due to the variety of applications and their importance, WSN will need to be connected to the Internet. We discuss the issues involved in this integration. In particular, we point out why all IP-sensor networks are infeasible, and suggest the feasible alternatives in the case of both homogeneous and heterogeneous networks. [1]

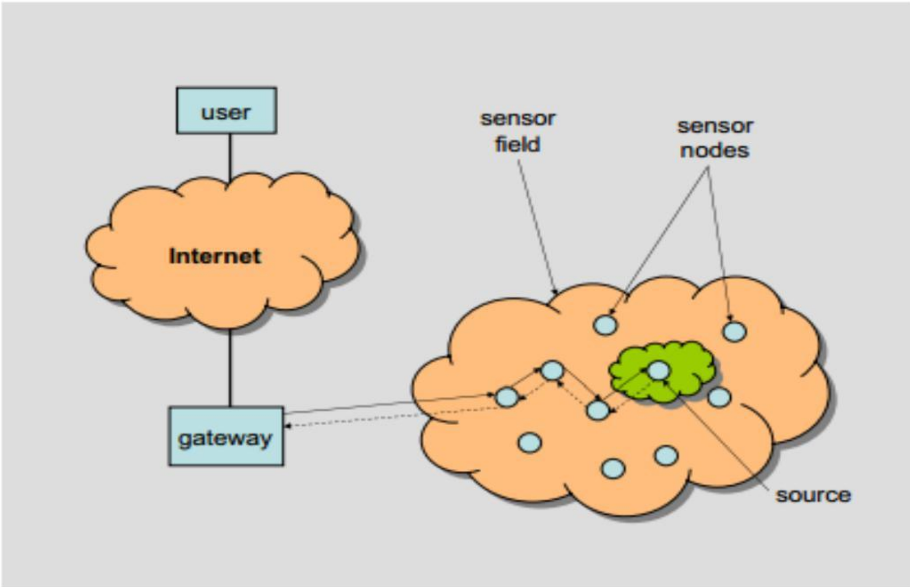

Fig 2: Communication architecture using a gateway. The gateway performs the necessary conversion of protocols including IP [1]

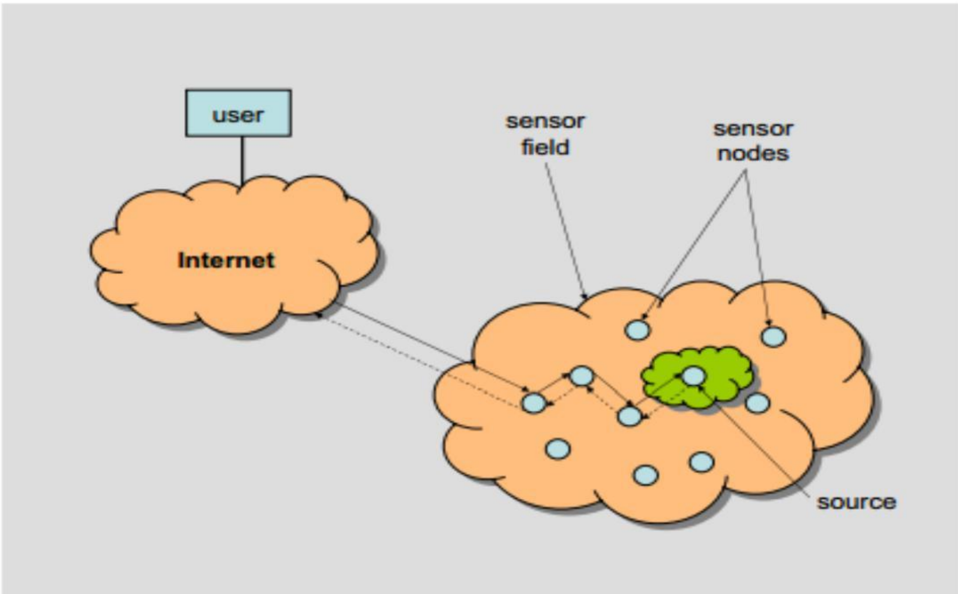

Fig 3: Communication Architecture with direct Connections [1]

II] In this report, we investigate the important requirements of communication architecture of wireless sensor networks for wide-area large scale soil moisture estimation and wetlands monitoring and explain the key issues that are faced in the design of the wireless sensor network monitoring strategy. We review the communication protocols and algorithms in MAC layer and network layer, and examine the standard components in the sensor network architecture. Based on the survey, we recommend the multi-hop and cluster based sensor network communication architecture for the proposed applications. We further study the MAC layer and network layer communication protocols for wireless sensor networks with the applications for wide area large scale soil moisture estimation and wetlands monitoring. [2]

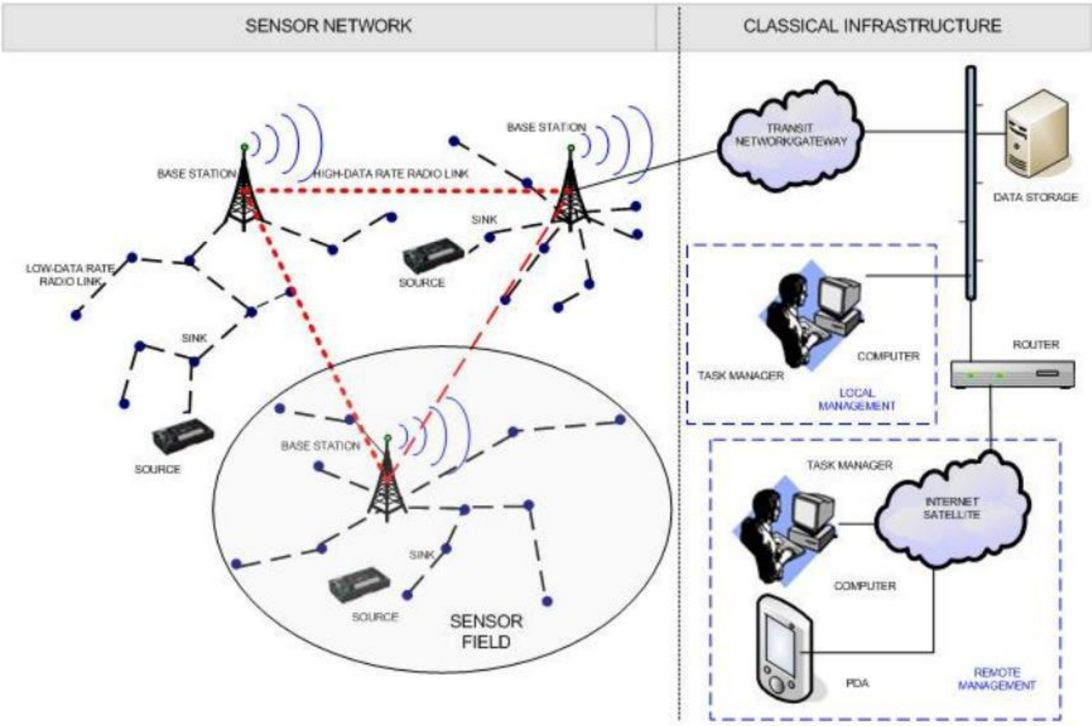

Fig 4: Sensor Network backbone Infrastructure. [II]

A basic sensor node typically comprises of five main components and they are namely controller, memory, sensors and actuators, communication device and power supply (see Figure below). A controller is to process all the relevant data, capable of executing arbitrary code. Memory is used to store programs and intermediate data. Sensors and actuators are the actual interface to the physical world. These devices observe or control physical parameters of the environment. The communication device sends and receives information over a wireless channel. And finally, the power supply is necessary to provide energy. In wireless sensor networks, power consumption efficiency is one of the most important design considerations. Therefore, these intertwined components have to operate and balance the trade-offs between as small energy consumption as possible and also the need to fulfil their tasks.

III] There are two fundamental deployment architectures for UW-ASNs, i.e., two dimensional architecture, where sensors are anchored to the bottom of the ocean, and the three-dimensional architecture, where sensors float at different ocean depths covering the entire monitored volume region. A three-dimensional deployment strategy is usually adopted when the phenomena cannot be adequately observed by bottom anchored sensor network. [3]

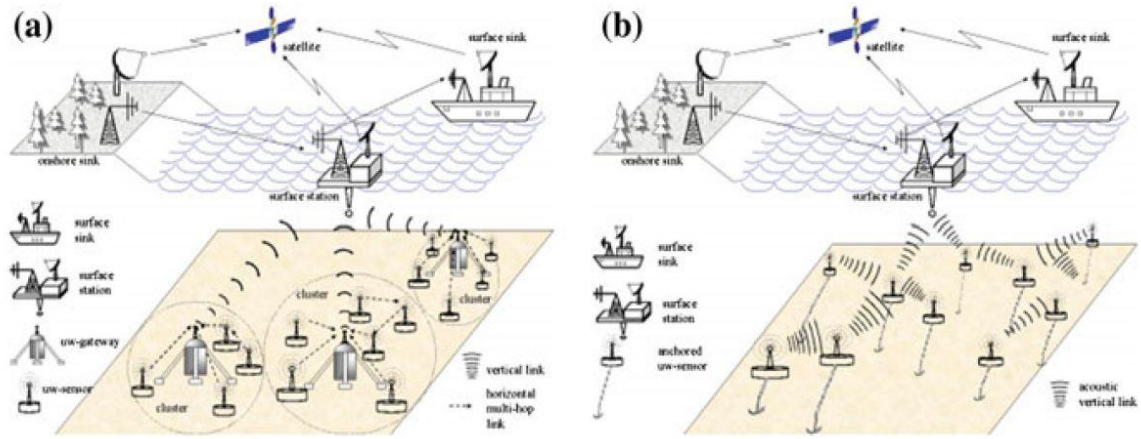

Fig 5: Underwater Deployment Strategies. [III]

UW-ASN network designer has following objectives to meet:

(1) Determine the minimum number of sensors to be deployed while not compromising
the target coverage (sensing) and connectivity.
(2) Provide guidelines on how to choose the optimal deployment surface area, given a target region.
(3) Study the robustness of sensor network to node failures, and include appropriate number of redundant sensors in the design to account for node failures.

The author proposes a 3D grid-based coordinate system in which the whole sensing volume is divided into cubes of size r. r is the resolution distance which is the edge length of a unit cube that sensor space is partitioned into. Coordination distance ‘a’ is also specified which indicates the distance in neighboring cubes for a node to coordinate its

3D Random	Simplest Strategy, No coordination from surface station. Sensors are randomly deployed, sensors choose their depth randomly. Each sensor informs its position	Will require greatest number of sensor nodes to cover a region. Might result in coverage-less patches. Only suitable for less critical applications
Bottom-Random	Sensors are randomly deployed at the bottom. Each sensor informs its location. Surface station then calculates the optimal depth for each sensor. Sensors then take that depth	Relatively more efficient strategy. Better 1-coverage probability is ensured
Bottom-Grid	Assisted by AUVs. Sensors deployed to pre-defined target locations to obtain grid deployment at the bottom of ocean. Each sensor then accordingly floats to its designated depth	The benefit of reduced number of sensors might be offset by expensive deployment strategy. Yields best result for 1-coverage

Fig 6: Node Deployment Strategies. [III]

depth. While a node arranges its depth, it exchanges information with the nodes within the coordination distance.

IV] Cloud computing focuses on delivery of reliable, secure, fault-tolerant, sustainable, and scalable infrastructures for hosting Internet-based application services. These applications have different composition, configuration, and deployment requirements. Quantifying the performance of scheduling and allocation policy on a Cloud infrastructure (hardware, software, services) for different application and service models under varying load, energy performance (power consumption, heat dissipation), and system size is an extremely challenging problem to tackle. To simplify this process, in this paper we propose CloudSim: a new generalized and extensible simulation framework that enables seamless modelling, simulation, and experimentation of emerging Cloud computing infrastructures and management services. The simulation framework has the following novel features: (i) support for modelling and instantiation of large scale Cloud computing infrastructure, including data centers on a single physical computing node and java virtual machine; (ii) a self-contained platform for modelling data centers, service brokers, scheduling, and allocations policies; (iii) availability of virtualization engine, which aids in creation and management of multiple, independent, and co-hosted virtualized services on a data center node; and (iv) flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services.
This paper also proposes a cloudsim architecture to run our cloud simulations on. [4]

CloudSim: A Toolkit for Modelling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms

A holistic software framework for modelling Cloud computing environments and performance testing application services. Clouds enable platform for dynamic and flexible application provisioning by exposing data center's capabilities as a network of virtual services. So users can access and deploy applications from anywhere in the Internet driven by demand and QoS requirements.

It's not possible to perform benchmarking experiments in repeatable, dependable, and scalable environment using real-world Cloud. Considering that none of the current distributed system simulators offer the environment that can be used for modelling Cloud, we present CloudSim.
Some Classes of Cloudsim are:
BwProvisioner: This is an abstract class that models the policy for provisioning of bandwidth
to VMs. The main role of this component is to undertake the allocation of network bandwidths to a set of competing VMs that are deployed across the data center. Cloud system developers and researchers can extend this class with their own policies (priority, QoS) to reflect the needs of their applications. The BwProvisioningSimple allows a VM to reserve as much bandwidth as required; however, this is constrained by the total available bandwidth of the host.
CloudCoordinator: This abstract class extends a Cloud-based data center to the federation. It is responsible for periodically monitoring the internal state of data center resources and based on that it undertakes dynamic load-shredding decisions. Concrete implementation of this component includes the specific sensors and the policy that should be followed during load-shredding. Monitoring of data center resources is performed by

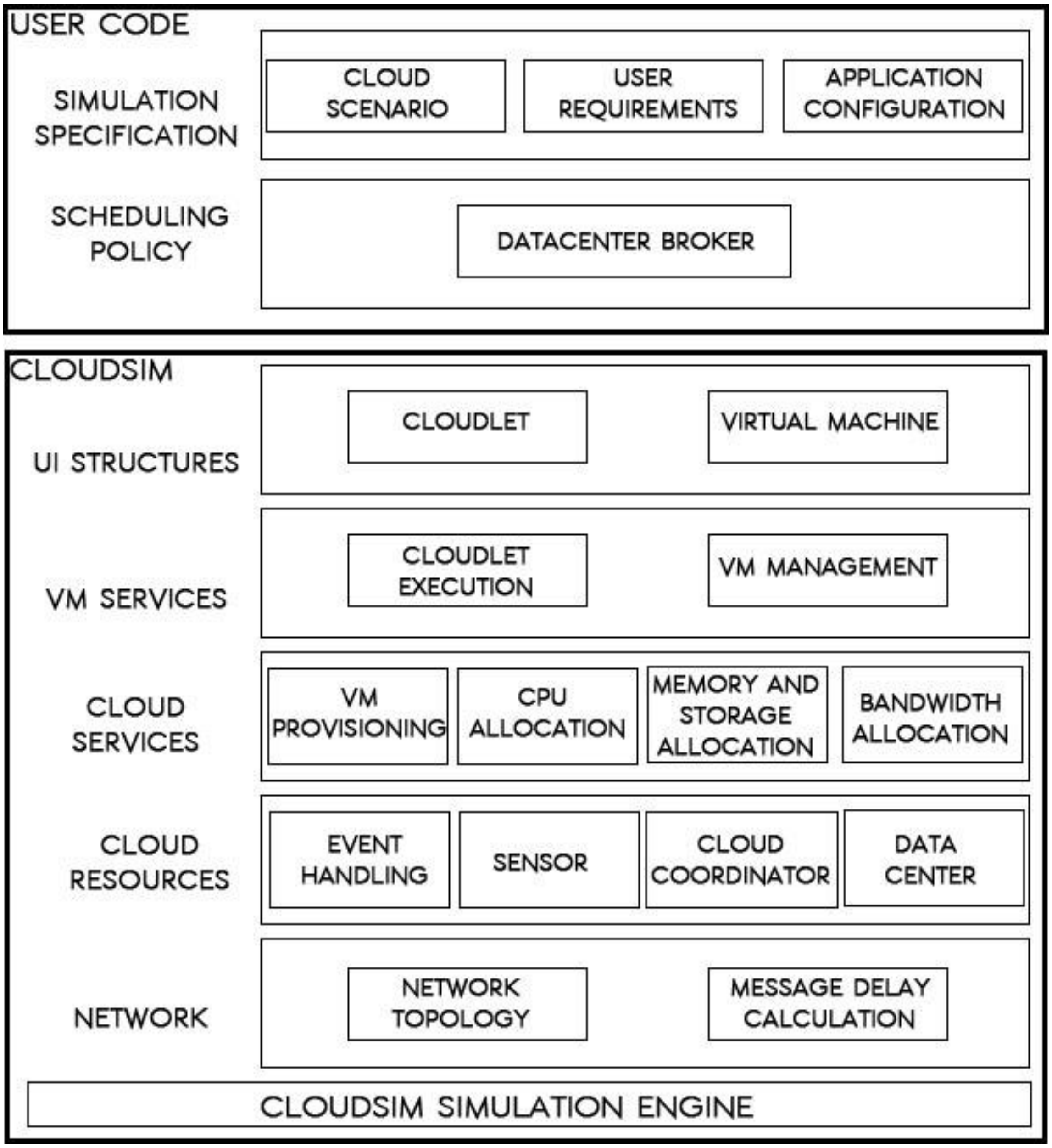

Fig 7: Layered CloudSim architecture [IV]

the updateDatacenter() method by sending queries Sensors. Service/Resource Discovery is realized in the setDatacenter()abstract method that can be extended for implementing custom protocols and mechanisms (multicast, broadcast, peer-to-peer). Further, this component can also be extended for simulating Cloud-based services such as the Amazon.

V] The system presents an integrated wireless sensor network (WSN) to monitor the information from agriculture systems namely temperature, humidity, pondus hydrogenii (pH) value…etc. The purpose is to provide a faster and more convenient platform for the client to obtain information from an array of sensor nodes that has been set-up in an agricultural system. A WSN will collect the values of various parameters from the front-end sensors at the host end. At the client sides, one can use the internet to request for Web Services that will store this big data into distributed SQL databases which are already in our proposed cloud system. In addition, this work presents the concept of cloud computing and services. The benefits of this system include basic computing hardware and reasonable storage capacities making it suitable for any smart device which can monitor real-time farmland information anywhere. The customers can fully access our cloud service using devices that have internet capabilities. [5]

VI] In a WSN-based marine environment monitoring system, various kinds of sensors are used to monitor and measure different physical and chemical parameters such as water temperature, pressure, wind direction, wind speed, salinity, turbidity, pH, oxygen density, and chlorophyll levels.

While the development and deployment of an adaptive, scalable and self-healing WSN system need to address a number of critical challenges such as autonomy, scalability, adaptability, self-healing and simplicity, the design and deployment of a lasting and scalable WSN for marine environment monitoring should take into account the following challenges different from those on land:

(1) Higher water resistance: Sensor nodes of a marine monitoring system require greater levels of water resistance;

(2) Stronger robustness: A marine monitoring system needs stronger robustness, since the marine environment with waves, marine currents, tides, typhoons, vessels, etc., is aggressive and complex, and causes movement of nodes;

(3) Higher energy consumption: Energy consumption is higher due to long communication distances and an environment in constant motion;

(4) More unstable line-of-sight: The oscillation of the radio antenna can cause a more unstable line-of-sight between transmitters and receivers.

(5) Other problems: There are also some other problems including the difficulty for deployment and maintenance of nodes, the need for buoy and mooring devices, sensor coverage problems and possible acts of vandalism.[6]

PROPOSED FRMEWORK

Figure shows the proposed wireless sensor network architecture for monitoring 3D environments, which consists of sensor nodes, sink nodes, a base station, a server and user terminals. Sensor nodes can sense and monitor the in-situ environmental parameters such as water temperature, salinity, turbidity, pH, oxygen density and chlorophyll levels, and transmit the collected data to sink nodes via wireless communication using ZigBee or some other communication protocol. Communication between sensor nodes and a sink node is usually point-to-point. A sink node collects data from a group of sensor nodes, and transmits the collected data to the base station. The server stores and processes the received data from the base station. The user terminals connect the server over the Internet. The Server here consists of cloud platform.

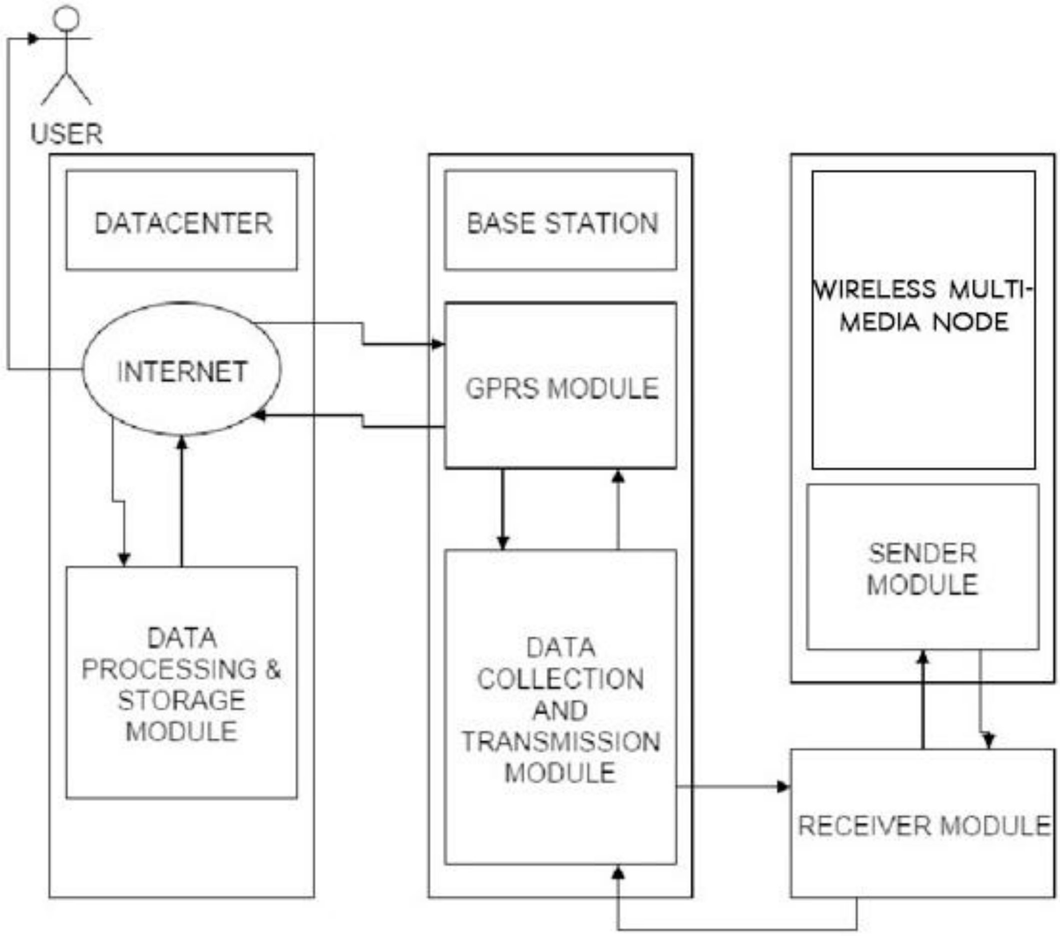

Fig 8: Proposed System Architecture

The design and deployment of a lasting and scalable WSN for marine environment monitoring should carefully take into account the following factors: the hostile environment, the network topology, communication protocols, the number of nodes, buoys, mooring systems, oceanographic sensors, energy supply, and so on.

General Sensor Node

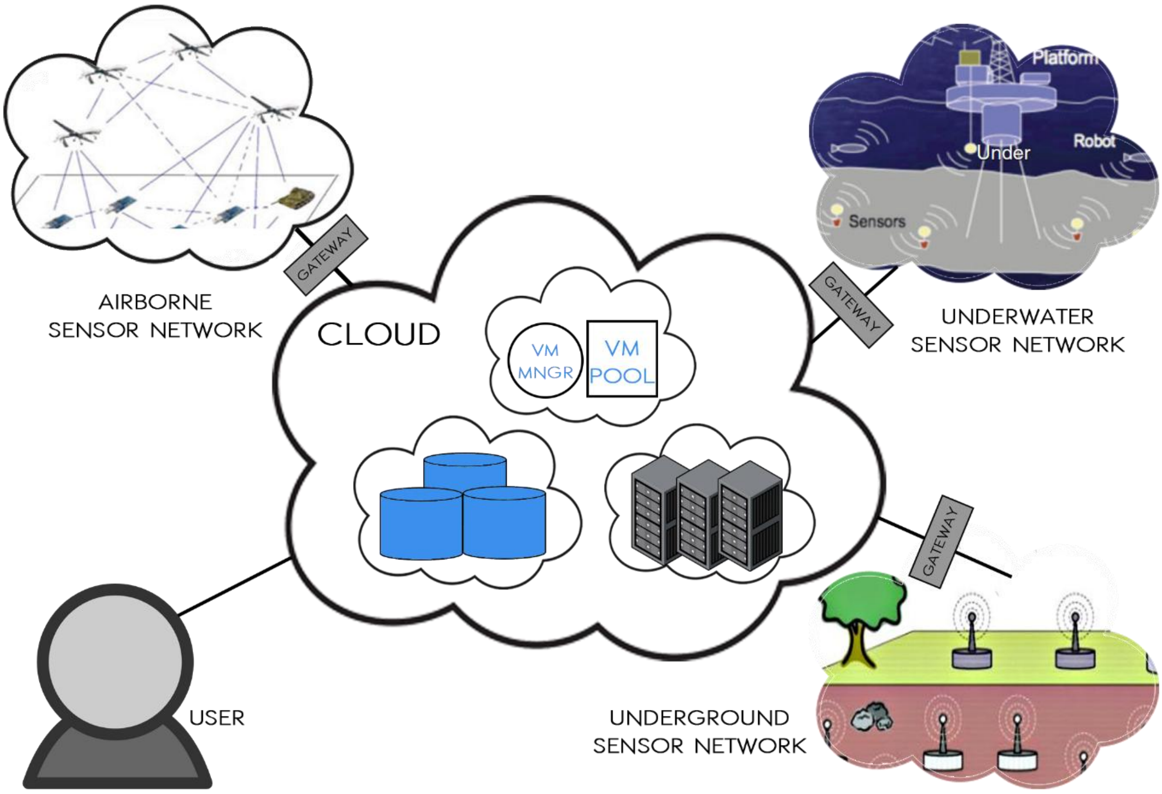

Fig 9: Common Architecture of 3D monitoring system

Figure below shows an architecture of a general sensor node in a marine environment monitoring system. It usually includes a buoy device in order to protect electronic devices of nodes against water. A marine monitoring sensor node normally consists of the following four main modules:

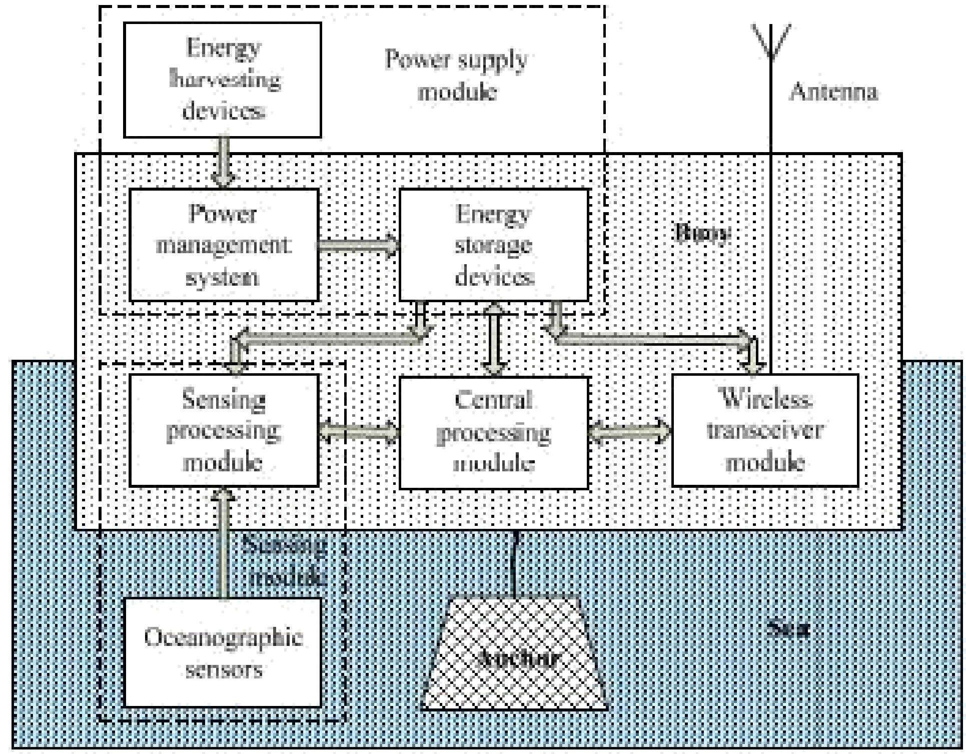

Fig 10: Common Architecture of 3D monitoring system

(1) A sensing module for data acquisition;

(2) A central processing module for local data processing and storage;

(3) A wireless transceiver module for wireless data communication;

(4) A power supply module for energy supply.

Wireless Communication Technologies

A sensor node normally incorporates a radio module for wireless communication. The transmitted distance of wireless communication can be anywhere between a few meters (Bluetooth, ZigBee, Wi-Fi, etc.) and thousands of kilometers (GSM or GPRS radio communication). Wireless communication has various standards and technologies including Bluetooth, ZigBee, Wi-Fi, GSM, GPRS and WiMAX Table 2 provides a summary and brief comparison of these communication technologies. Usually, two or more wireless communication technologies are used in a real wireless sensor network. In particular, underwater acoustic communication technologies can be a good choice for data collection and exchange among underwater sensors. Table 1 provides a summary and brief comparison of these communication technologies. Usually, two or more wireless communication technologies are used in a real wireless sensor network. In particular, underwater acoustic communication technologies can be a good choice for data collection and exchange among underwater sensors. Generally, the longer the range a radio module must transmit, the more energy consumption a radio module will have. The choice of a wireless communication technology depends on the amount and frequency of the transmitted data, transmission distance, and amount of available energy.

Technology	Standard	Description	Throughput	Range	Frequency
WiFi	IEEE 802.11a; 802.11b/g/n	System of wireless data transmission over computational networks.	11/54/300 Mbps	<100 m	5.8 GHz 2.4 GHz
Bluetooth	IEEE 802.15.1	Industrial specification for WPAN which enables voice and data transmission between different devices by means of a secure, globally free radio link (2.4 GHz).	v. 1.2: 1 Mbps v. 2.0: 3 Mbps UWB: 53–480 Mbps	Class 1: 100 m Class 2: 15–20 m Class 3: 1 m	2.4 GHz
ZigBee	IEEE 802.15.4	Specification of a set of high-level wireless communication protocols for use with low consumption digital radios, based on WPAN standard IEEE 802.15.4.	250 Kbps	<75 m	2.4 GHz
WiMAX	IEEE 802.16	Standard for data transmission using radio waves.	<75 Mbps	<10 km	2–11 GHz
GSM		Standard system for communication via mobile telephones incorporating digital technology	9.6 Kbps	Dependent on service provider	850/900/1800 /1900 MHz
GPRS		GSM extension for unswitched (or packaged) data transmission.	56–144 Kbps	Dependent on service provider	850/900/1800 /1900 MHz

Table 1: Various communication techniques

Sensing Parameters and Sensors

The operating principle of sensors is to respond to changes in their environment by producing an electrical signal in the form of voltage, current, or frequency [31]. Sensors can commonly be divided into physical sensors and chemical sensors. In a marine monitoring system, physical sensors are used to measure some physical parameters, such as temperature, humidity, pressure, wind speed and wind direction, and chemical sensors are used to sense various chemical parameters (salinity, turbidity, pH, nitrate, chlorophyll, dissolved oxygen (DO), etc.) as shown in Table.

Sensors	Monitoring Parameters	Range	Accuracy	Power Supply	Unit	Manufacturer
SBE 16plus V2	Temperature	−5 to +35 °C	±0.005 °C	9–28 V	°C	Sea-Bird Electronics
GT301	Pressure	0 to 60	< ±0.5% of FRO	24 V	bar	Kongsberg Maritime
SBE 16plus V2	Conductivity (Salinity)	0–9	± 0.0005	9–28 V	S/m	Sea-Bird Electronics
OBS-3+	Turbidity	Mud: 5000–10,000 mg/L Sand: 50,000–100,000 mg/L	0.5 NTU	15 V	NTU	Campbell Scientic
PS-2102	pH	0 to 14 pH	±0.1	N/A	pH	PASCO
YSI 6025	Chlorophyll	0 to 400 μg/L	0.1 μg/L	6 V	μg/L	YSI
ISUS V3	Nitrate	0.007 to 28 mg/L	±0.028 mg/L	6–18 V	mg/L	Satlantic
SBE 63	Dissolved oxygen (DO)	120% of surface saturation in all natural waters	0.1	6–24 V; 35 mA	mg/L	Sea-Bird Electronics

Table 2: Various sensors available

Underwater Wireless Sensor Network Architecture

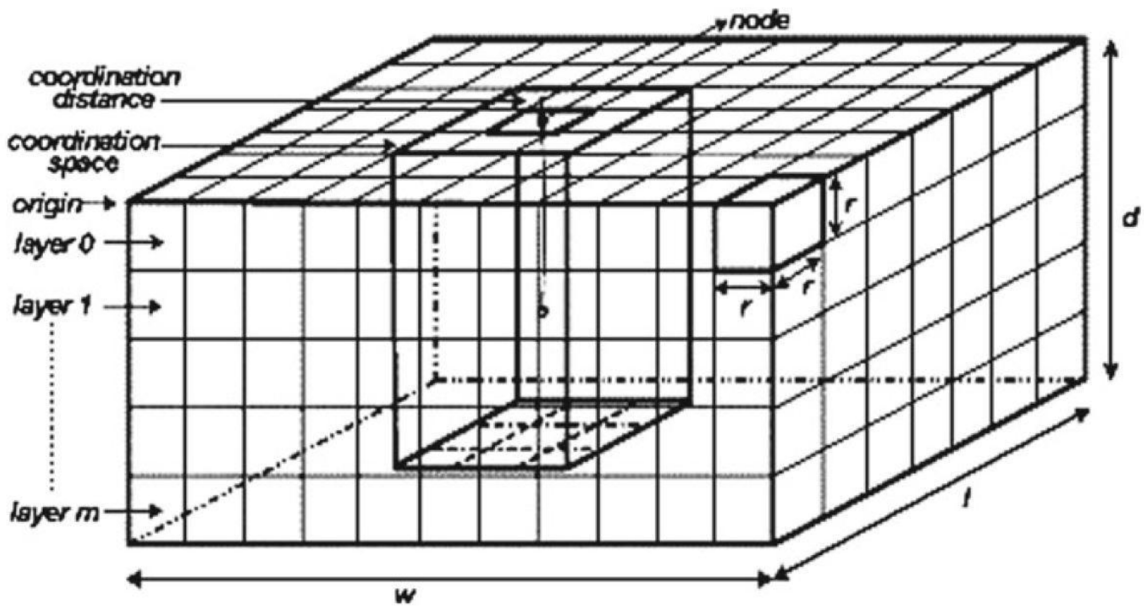

Fig 11: Underwater Scenario Model

Here for the purpose of our study we consider a cube shaped underwater environment which is divided into grids on which nodes are deployed as per various deployment schemes. In our simulation we use bottom grid deployment scheme.

Grid-Based Deployment Scheme

Here we use a 3D grid-based coordinate system in which the whole sensing volume is divided into cubes of size r. r is the resolution distance which is the edge length of a unit cube that sensor space is partitioned into. Coordination distance ‘a’ is also specified which indicates the distance in neighboring cubes for a node to coordinate its depth. While a node arranges its depth, it exchanges information with the nodes within the coordination distance.

Effects of Oceanic Parameters

It is paramount to have reasonable understanding of behavior and characteristics of acoustic signals when designing3Dsubmerged network. The effect of ocean parameters on acoustic signals in the context node location estimation. Due to acoustic nature of the communication signals, the distance measurements in UW-ASNs can be affected by temperature, salinity and depth. Leroy et al. proposed an equation for sound speed which is a function of temperature (T), salinity(S), depth (Z) and latitude (θ) in all oceans and open seas.

Due to special mechanical properties of sea water, sound moves at the mean Speed of 1500 m/s. It has to be said that network designers should keep these oceanic Parameters in mind while deciding on network latency, connectivity, and deployment and configuration algorithms.

Mobility of Nodes

The movement of underwater objects is closely related to many environment factors such as water current and water temperature [8]~[10]. In different environments, the mobility characteristics of underwater objects are different. For example, the mobility patterns of objects near the seashore demonstrate a certain semi-periodic property because of tides; but for objects in rivers, their moving behaviors have no such property. So it is almost impossible to devise a generic mobility model for underwater objects in all environment conditions. Here we design a three-dimensional random walk model to depict the mobility of underwater sensor nodes. Assume the generation interval of the mobility event is D, the simulation process can be given:

Algorithm – a three-dimensional random walk algorithm

```
while (SimTime>0){
for (each ordinary sensor node si (xi, yi, zi)∈S) {
// In each mobility event, the node shifts with probability pe
// and keeps static with probability 1-pe
if (random(0, 1) < pe) {
//random walks process
xi=xi+randomInt(0, mx)× [2×(random(0, 1)<pdi_x)-1];
yi=yi+randomInt(0, my)×[2×(random(0, 1)<pdi_y)-1];
zi=zi+randomInt(0, mz)×[2×(random(0, 1)<pdi_z)-1];
}
```

}
SimTime=SimTime-D;
}

Here the function randomInt(a, b) and random (a, b) respectively generate a random integer and a random real number between a and b. In the random walk process, take the x direction as an example, mx is the possible maximum offset in x direction when nodes shift, and the shifting node jumps up with probability pdi_x and down with probability 1-pdi_x in x direction, random (0, 1)<pdi_x is a Boolean expression, return 1 when it is true and return 0 when it is false.

Routing Protocols for UWSN:

Routing protocols for WSNs can be classified into flat, location-based and hierarchical/cluster categories. The flat structure can be considered a suitable solution for many IoT applications (in network with homogeneous nodes), such as comfortable homes and offices, healthcare, environmental monitoring, and many smart city services. Many applications employed in these scenarios have low tolerance for packet delay and loss. Routing protocols based on clustering are an alternative to improve QoS and energy consumption fora set of IoT applications [17], such as multimedia-based fire detection [18]. A hierarchical architecture has nodes with different roles or functionalities (heterogeneous nodes, which can be classified as cluster-head and non-head nodes). They work based on clustering sensor nodes, where the nodes of the cluster communicate with each other (sensor-to-sensor), and mainly with the leader node (cluster-head),responsible for communicating outside the cluster (sensor-to-BS). Some nodes (cluster-head) can have cameras and extra batteries. There are many algorithms for the election of cluster-head, which analyze the key features as follows: residual energy, link quality and location.

REL: REL proposes an end-to-end route selection scheme based on cross-layer information with a minimum overhead. To achieve energy-efficiency, the nodes send their residual energy to neighboring nodes by means of an on-demand piggyback scheme. Additionally, REL uses an event-driven mechanism to provide load balancing and avoid the energy hole problem. REL uses the link quality of wireless links and residual energy during the routing selection process to increase the system's reliability and assure QoS support for IoT applications. Additionally, it includes an event-driven mechanism to provide load balancing and avoid the premature death of nodes/networks.

AODV: On demand Distance Vector) protocol [23], which was originally proposed in RFC 3965. The popularity of AODV is due to its well-defined structure and low complexity. In AODV, on-demand routes can be discovered, which decrease the overhead, by using pairs of Route Request (RREQ) and Route Reply (RREP) messages. However, the route selection process is only carried out on the basis of the minimal number of hops, which is not suitable for ensuring energy-efficiency and reliable data transmission. This can be explained by the fact that in many cases, a short route, in terms of hops, can be more susceptible to packet loss, due to both noise and interference that affect the link quality. The lack of energy-efficiency mechanism results in energy holes and an uneven distribution of scarce network resources. Moreover,AODV only stores one possible route for a given destination node. This means that if a single route fails or is unavailable, a new route must be discovered, which requires more time and increases the delay or failure rate of data delivery.

LABILE: LABILE proposes a routing algorithm based on lexical structures and link quality evaluation. Through the use of LQI, i.e., a metric provided by the physical layer of IEEE 802.15.4 standard, LABILE is able to evaluate the link quality. The LABILE proposal evaluates end-to-end link quality, by classifying the possible values of LQI into good or bad. In specific terms, it determines a threshold value for link classification, where the lowest values of LQI (below the threshold) are considered bad, and represent links that are more susceptible to packet loss. During the route discovery process, all the bad links are counted, recorded and reported with the aid of an additional field in RREQ and RREP messages, which is calledWeakLinks. The purpose of LABILE is to select routes with good link qualities. However, this behavior implies that these routes have an exhaustive use, and lead to the premature death of these nodes. This is due to a lack of mechanisms for determining when there is a need to use alternative routes. Thus, LABILE does not take account energy-efficiency into account during the route selection process and load balancing mechanisms as expected for multipath WSN/IoT scenarios.

Cloud Server Architecture:

The cloud server employs the concept of distributed computing with virtualization to improve its computational power. The idea of distributed computing is to divide the whole work load into smaller units. Each work fragment will be given to a corresponding slave computer that will do the computing and then will send back the results to the master computer. Through this technique, cloud computing can achieve a similar processing power to that of a “Super computer” with relatively lower cost and lesser fabrication complexity. In the past, our computing power was based on personal computers locally, but through virtualization technology we can now use cloud computing power which is centralized. The proposed cloud system consists of a master server computer and four virtual slave server computers. The data
are distributed onto the respective storage spaces of the four slave server computers which shall be used in the distributed computing process.

1] Datacenter is composed of a set of hosts and it is responsible for managing virtual machines (VMs) (e.g., VM provisioning). It behaves like an IaaS provider by receiving requests for VMs from brokers and creating the VMs in hosts.

2] DatacenterBroker- This class represents a broker acting on behalf of a user. It modifies two mechanisms: the mechanism for submitting VM provisioning requests to data centers and the mechanism for submitting the tasks to VMs. The CloudSim users have to extend this class for conducting experiments with their own policies.

3] Host-Host executes actions related to management of VMs (e.g., creation and destruction) and update task processing to VMs. A host has a defined policy for provisioning memory, processing elements, and bandwidth to virtual machines. A host is associated to a datacenter. It can host virtual machines.

4] VM-It represents a software implementation of a machine that executes applications called virtual machine (VM) which works like a physical machine. Each virtual machine divides the resources received from the host among tasks running on it.

5] Cloudlet-A cloudlet class is also known as a task. CloudSim represents the complexity of an application in terms of its computational requirements. This class is managed by the scheduling policy which is implemented in DatacenterBroker Class.

The proposed system consists of two parts WSN sensor nodes for gathering and transmitting data and the cloud datacentre for receiving data and processing it to give output to the user.

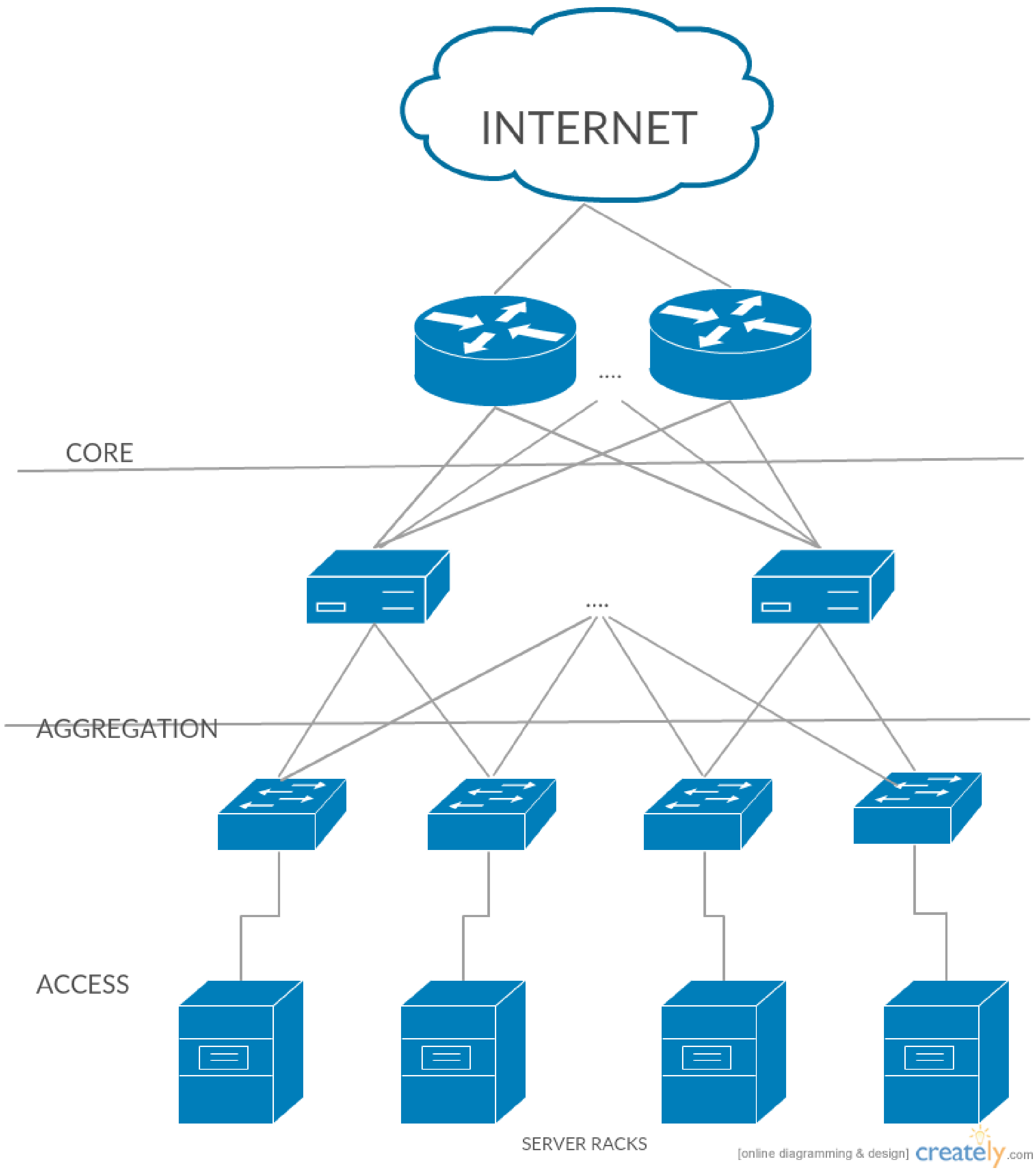

Fig 12: Cloud System Structure

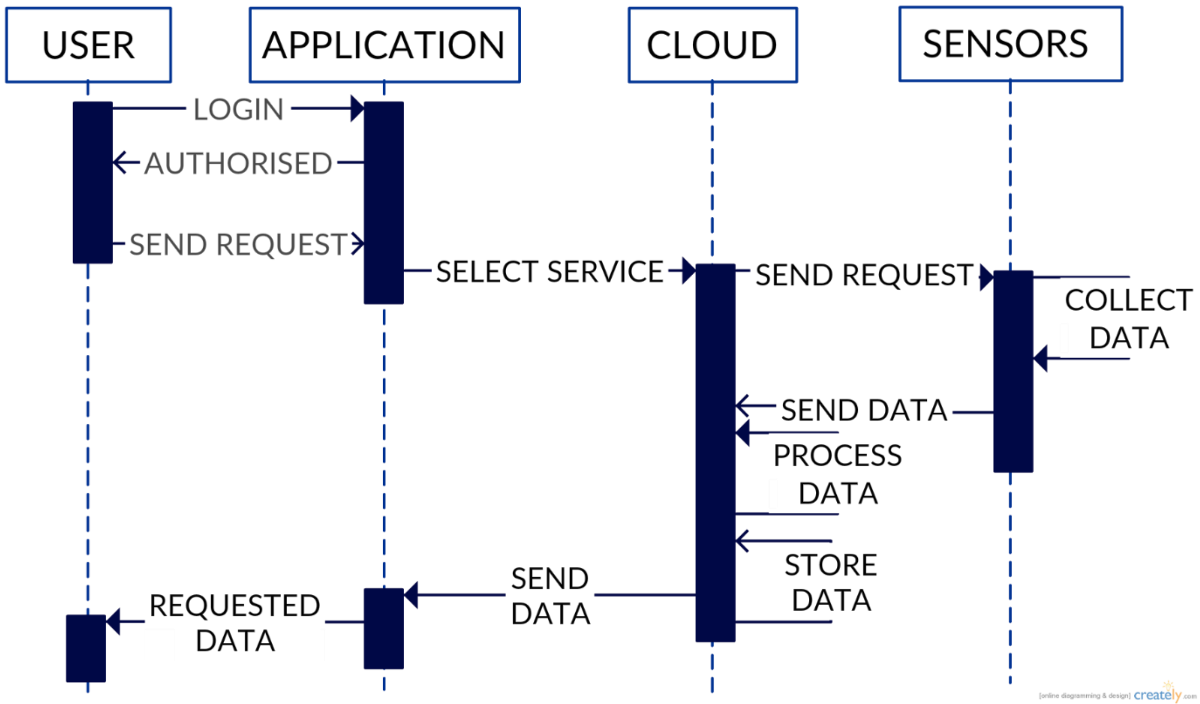

Fig 13: Sequence Diagram

DVFS in CloudSim:

The core features for DVFS simulation were added to a new package, called DVFS. In this package, governors of the five modes of DVFS, as they are present in the Linux kernel.

Their role is to determine if the frequency of the CPU must be changed and their decision is directly related to their intrinsic decision rule. In the simulator, a frequency change directly impacts the capability of CPUs, measured in MIPS. The use of DVFS directly affects the performance of the CPU capacity (and hosts), which are subject to regular changes during simulations.

This also involves changing the way the simulator handles the placement and management of virtual machines. For example, if the system decides to reduce the frequency, the sum of capacities of all virtual machines (VMs) may temporarily exceed the maximum capacity of the host. In this case, the size of each VM must temporarily be adjusted downward in proportion to the capacity of the new host. The same situation occurs when one or more new virtual machines have to be added to a host already in use. If the sum of capacities of the virtual machines already running in the host plus the capacities of new virtual machines created exceeds the capacity of the host, the size of all virtual machines has to be decreased before adding the new virtual machines to the host.

Also, when part of the capacity of a host is released, capacities of virtual machines still active may increase. This event occurs when a virtual machine has finished its execution or when the CPU frequency is increased. In this case, the capacity of all the remaining virtual machines are scaled up regarding the free capacity of the host, while taking care to do not exceed its maximum capacity.

When a broker is instantiated, it can contain the specification of a post-event that triggers a new event which starts the execution of a new sequence of cloudlets. The last change in the simulator is the energy model. Indeed, the energy model used depends on the host to be modelled in the simulation. This scenario is modelled in CloudSim by running a set of brokers, each containing one or several VMs in which cloudlets are executed. A broker may contain a post-event, which starts when all the VMs contained in it have finished their execution. By defining different types of VMs (with different MIPS) and different types of cloudlets (executing different number of instructions), it is possible to increase and manage the CPU load accurately. The size of a VM de- fines the percentage increase to the CPU load and the size of the cloudlet defines the execution time of the VM. So, this scenario is created by a set of brokers that launch a series of VMs and cloudlet pairs. Finally, to have the same interval sampling rate as in the Linux kernel, the simulation time interval is set to 10 ms.

The core hardware infrastructure services related to the Clouds are modelled in the simulator by a Datacenter component for handling service requests. These requests are application elements sandboxed within VMs, which need to be allocated a share of processing power on Datacenter's host components. By VM processing, we mean set of operations related to VM life cycle: provisioning of a host to a VM, VM creation, VM destruction, and VM migration.

A datacenter is composed of a set of hosts, which during their lifecycles work on the basis of managing the virtual machines. Host is a component that represents a physical computing node in a Cloud: it is assigned a pre-configured processing (expressed in millions of instructions per second – MIPS, per CPU core), memory, storage, and a scheduling policy for allocating processing cores to virtual machines.

Allocation of application-specific VMs to Hosts in a Cloud-based data center is the responsibility of the Virtual Machine Provisioner component.

This component exposes a number of custom methods for researchers, which aids in implementation of new VM provisioning policies based on optimization goals. . The default

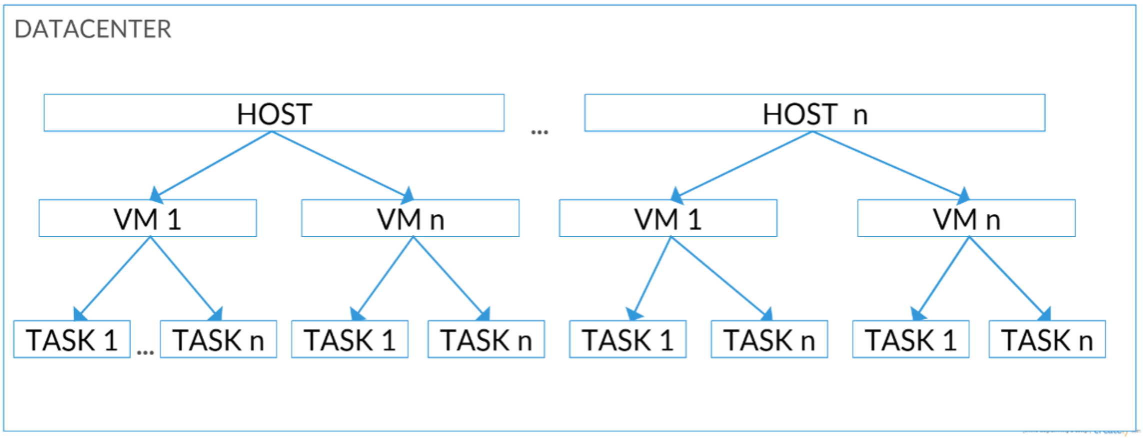

Fig 14: Host to VM Mapping

policy implemented by the VM Provisioner is a straightforward policy that allocates a VM to the Host in First-Come-First-Serve (FCFS) basis.

For each Host component, the allocation of processing cores to VMs is done based on a host allocation. The policy takes into account how many processing cores will be delegated to each VM, and how much of the processing core's capacity will effectively be attributed for a given VM. So, it is possible to assign specific CPU cores to specific VMs (a space-shared policy) or to dynamically distribute the capacity of a core among VMs (time-shared policy), and to assign cores to VMs on demand, or to specify other policies.

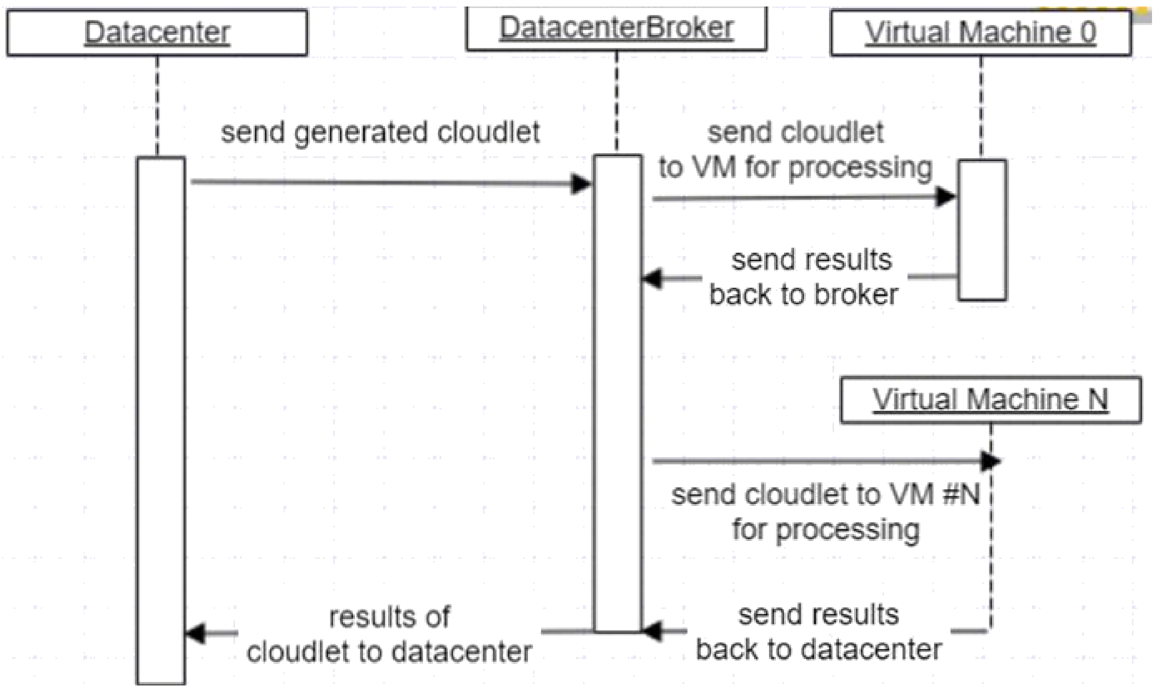

Fig 15: Class Design Diagram of CloudSim

As each of the components work in accordance to one another we define them as entities. The execution of these entities is done through threads which are allocated through the SimJava.

In the beginning of the simulation, each Datacenter entity registers itself with the CIS (Cloud Information Service) Registry. CIS provides database level match-making services for mapping user requests to suitable Cloud providers. Brokers acting on behalf of users consult the CIS service about the list of Clouds who offer infrastructure services matching user's application requirements. In case the match occurs the broker deploys the application with the Cloud that was suggested by the CIS.

Datacenter and VM architecture

At each level, CloudSim implements the time-shared and space-shared provisioning policies. To clearly illustrate the difference between these policies and their effect on the application service performance, in Figure we show a simple VM provisioning scenario. In this figure, a host with two CPU cores receives request for hosting two VMs, such that each one requires two cores and plans to host four tasks' units. More specifically, tasks t1, t2, t3, and t4 to be hosted in VM1, whereas t5, t6, t7, and t8 to be hosted in VM2.

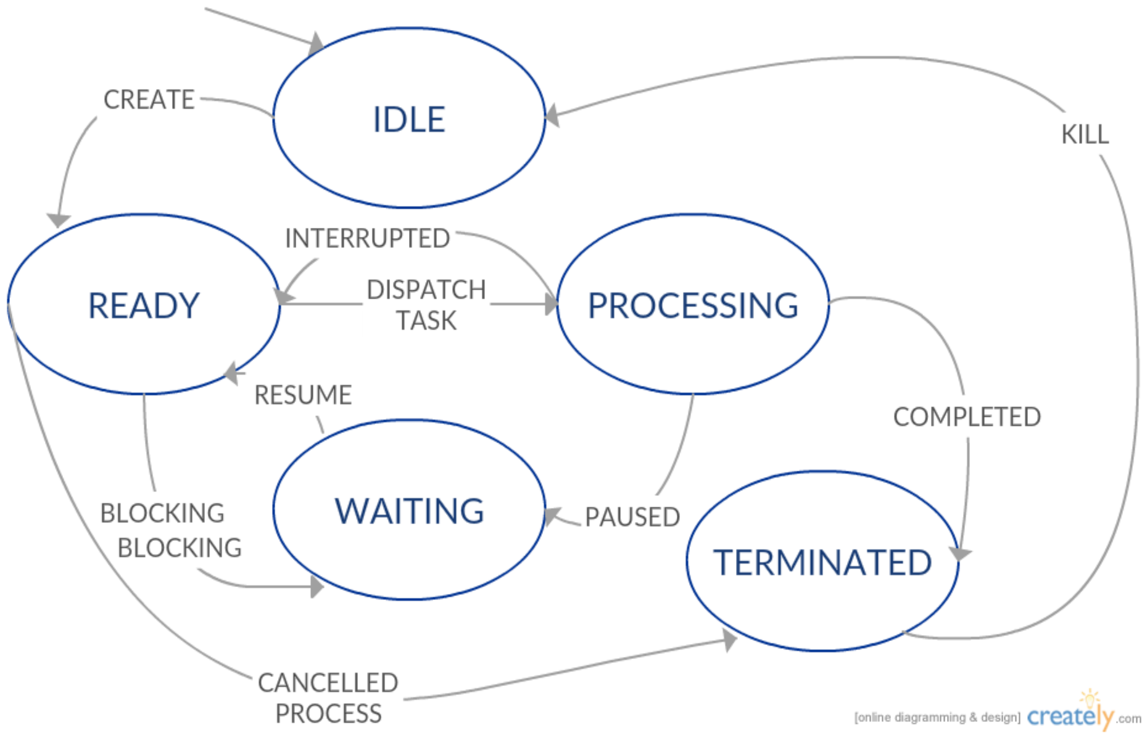

Fig 16: VM State Transition Diagram

Figure below presents a provisioning scenario, where the space-shared policy is applied to both VMs and task units. As each VM requires two cores, in space-shared mode only one VM can run at a given instance of time. Therefore, VM2 can only be assigned the core once VM1 finishes the execution of task units. The same happens for provisioning tasks within the VM1: since each task unit demands only one core, therefore both of them can run simultaneously. During this period, the remaining tasks (2 and 3) wait in the execution queue. By using a space-shared policy, the estimated finish time of a task p managed by a VM i is given by

eft(p)=est + rl capacity×cores(p)

where est(p) is the Cloudlet- (cloud task) estimated start time and rl is the total number of instructions that the Cloudlet will need to execute on a processor

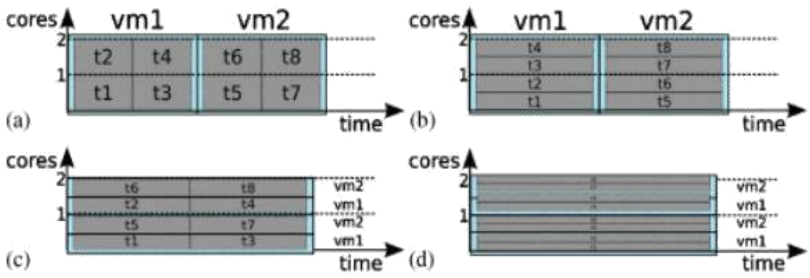

Fig 17: Effects of Different Provisioning Polices on task unit execution: a) Space-Shared Provisioning for VMs and tasks; b) Space-Shared provisioning for VMs and Time-Shared provisioning for tasks; c) Time-Shared provisioning for VMs and Shape-Shared provisioning for tasks; and d) Time-Shared Provisioning for VMs and tasks

The estimated time depends on the position of the Cloudlet in the execution queue, because the processing unit is used exclusively (space-shared mode) by the Cloudlet. Cloudlets are put in the queue when there are free processing cores available that can be assigned to the VM.

Modeling comprehensive network topologies to connect simulated Cloud computing entities (hosts, storage, end-users) is an important consideration because latency messages directly affect the overall service satisfaction experience. An end-user or a SaaS provider consumer who is not satisfied with the delivered QoS is likely to switch his/her Cloud provider; hence, it is a very important requirement that Cloud system simulation frameworks provide facilities for modeling realistic networking topologies and models.

The topology description is stored in BRITE format that contains a number of network nodes, which may be greater than the number of simulated nodes. These nodes represent various CloudSim entities including hosts, data centers, Cloud Brokers etc. This BRITE information is loaded every time CloudSim is initialized and is used for generating latency matrix.

PERFORMANCE ANALYSIS

Cloud Simulation Environment

Description	*Parameter*
System Architecture	x86
VM allocation Policy	DVFS
Workload	Random workload
No-of Users	2
No of Datacenters	2
VM Description	
MIPS	250
Size	10000MB
RAM	512MB
No of CPU	1

VM Performance Analysis:

Host1: 1000 MIPS. VM1: 100 MIPS, Cloudlet1: 150 Millions of Instructions.

VM2: 20 MIPS, Cloudlet 2: 80 Millions of Instructions.

1. DVFS enabled + VM live migration if host utilization above threshold

Comparison of Time shared and Space shared cloudlet allocation policies on the basis of throughput and avg. waiting time:

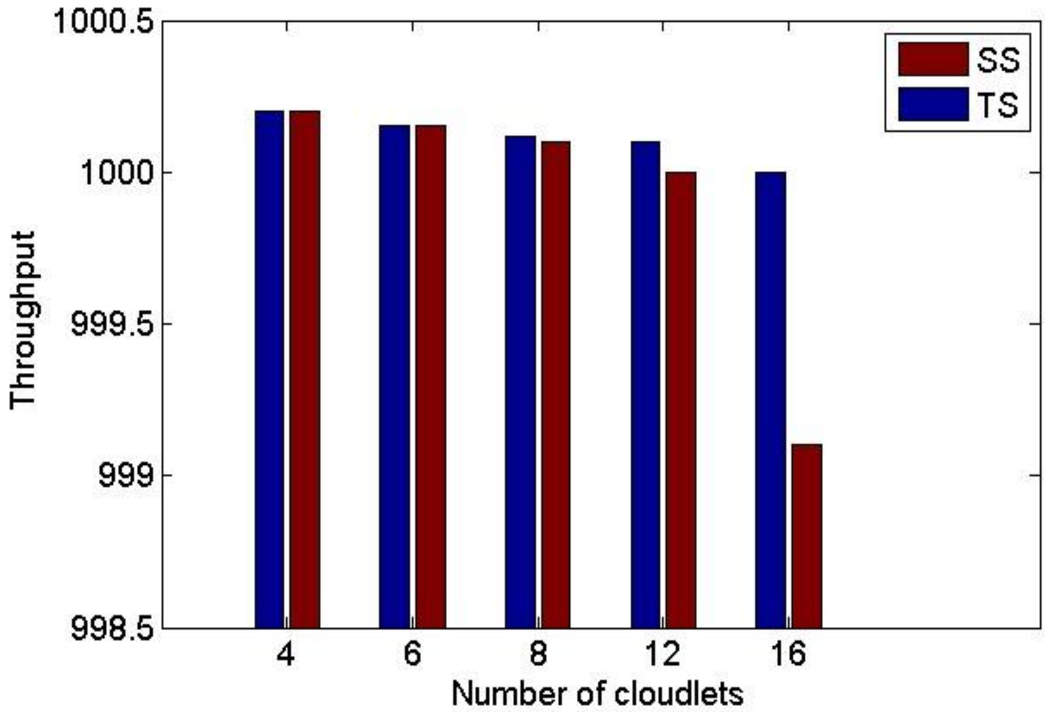

Fig 18: Throughput with 2 Cores

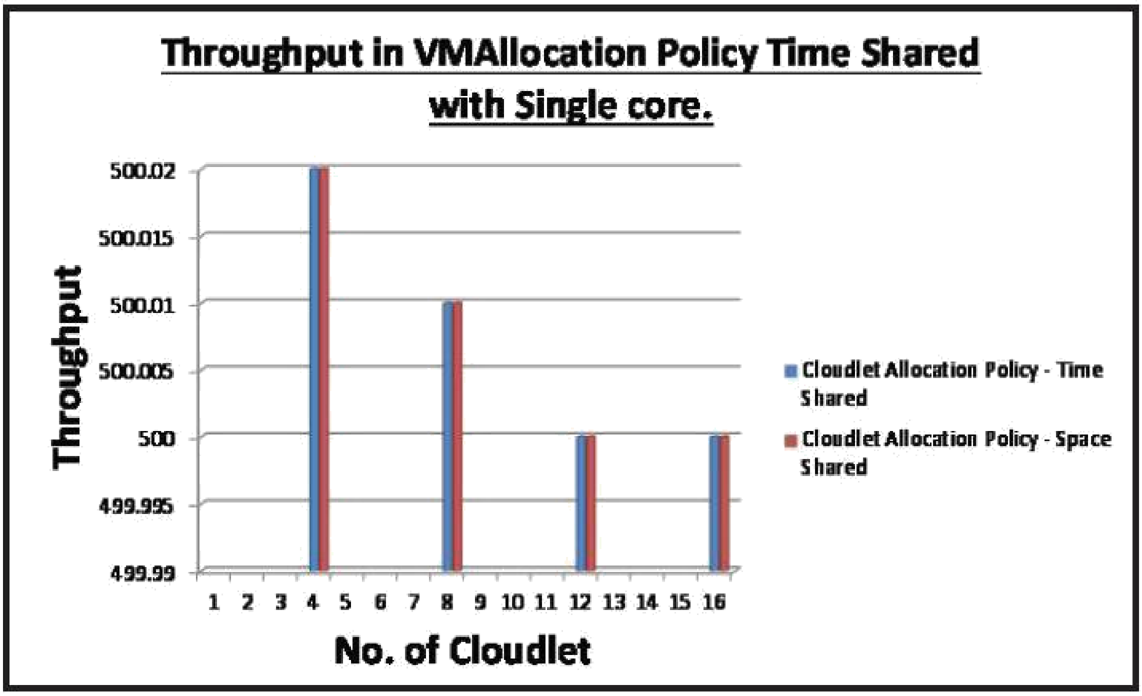

Fig 19: Throughput in Time Shared VMAllocation Policy

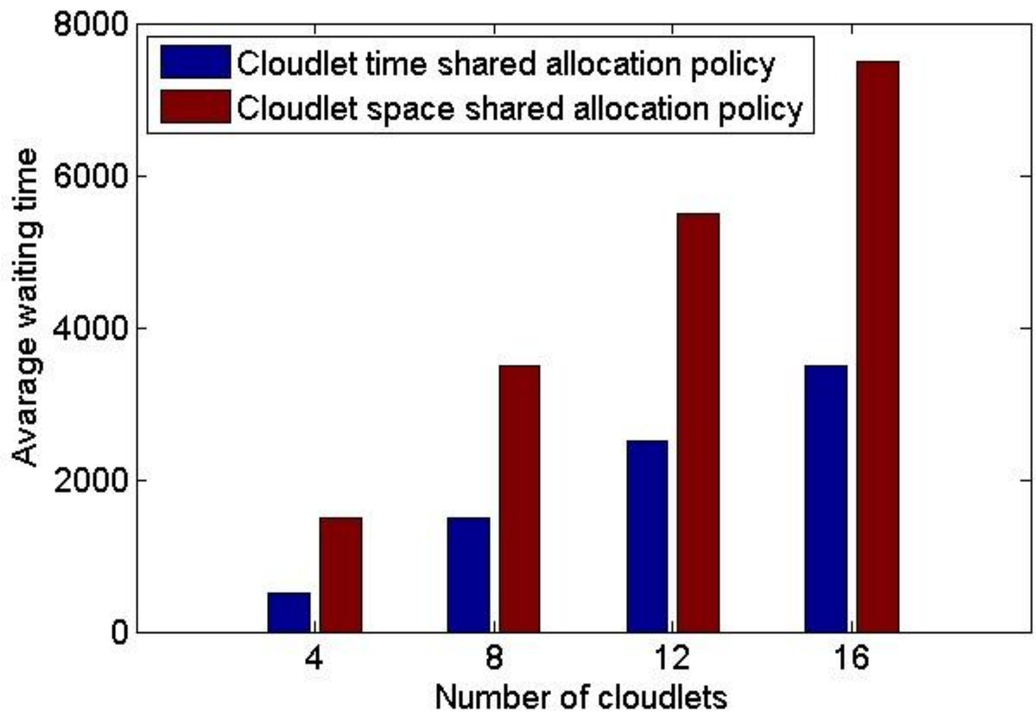

Fig 20: Avg Waiting Time with 2 Cores

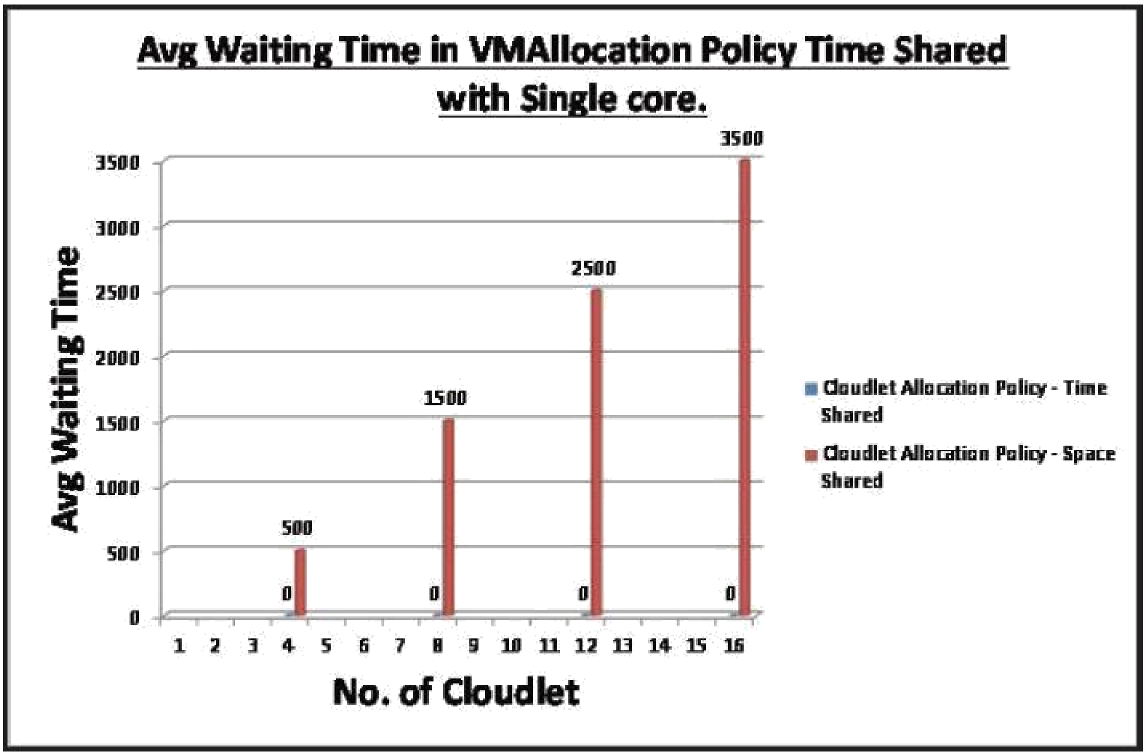

Fig 21: Avg Waiting Time in Time Shared VMAllocation Policy

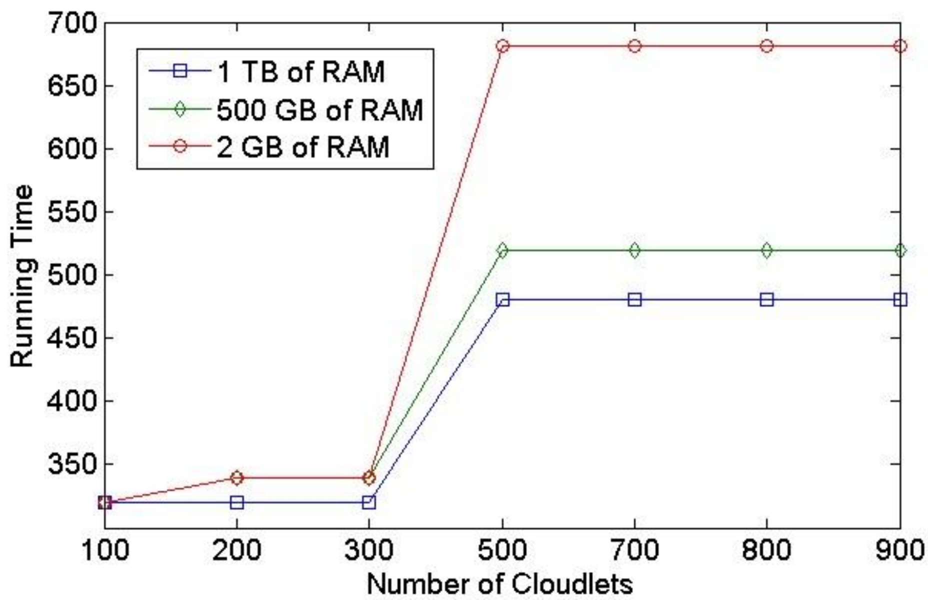

Fig 22: Effect of varying Ram

OBSERVATIONS:

- Here we see that time shared policy gives better results as compared to space shared policy due to having a lower average wait time and higher throughput.
- We also see that with the increase in RAM the running time of the processes decreases significantly.

UWSN SIMULATION ENVIRONMENT

Underwater Environment Parameters
Volume of the scenario- 10m x 10m x 10m
No of Sensor Nodes- 100
Mobility Model- Random Walk
Deployment- Bottom Grid Deployment

AODV Performance Analysis:

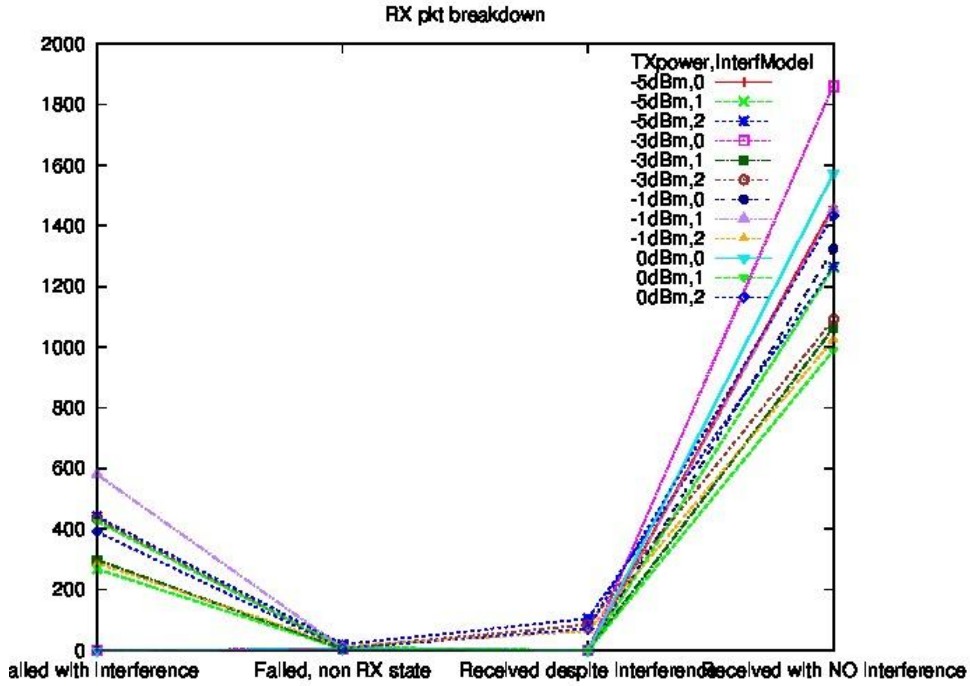

Fig 23: Received Packet Breakdown

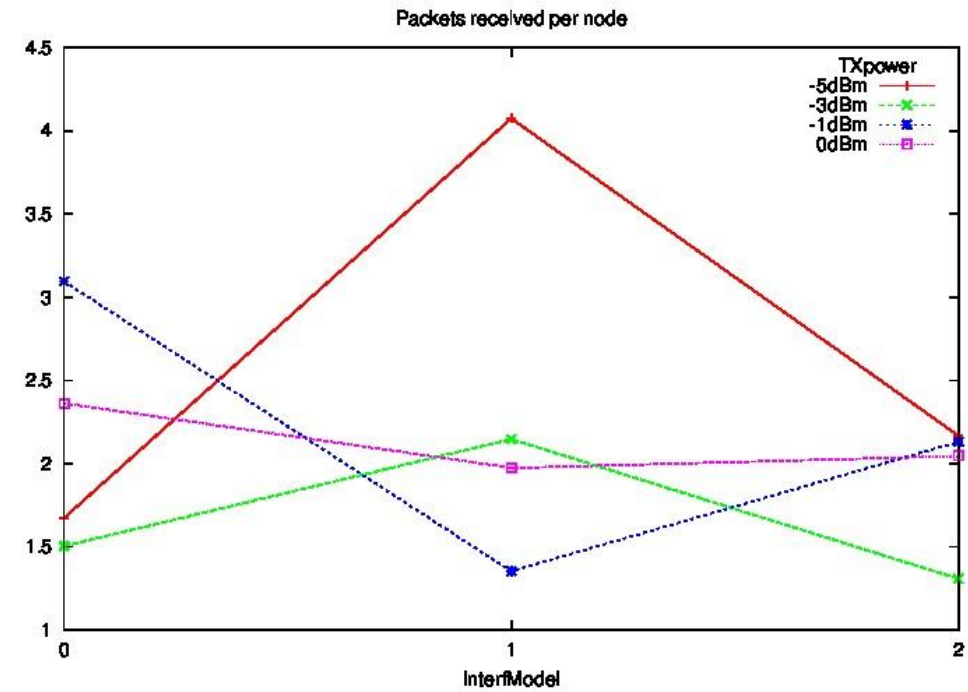

Fig 24: Packet Received per Node

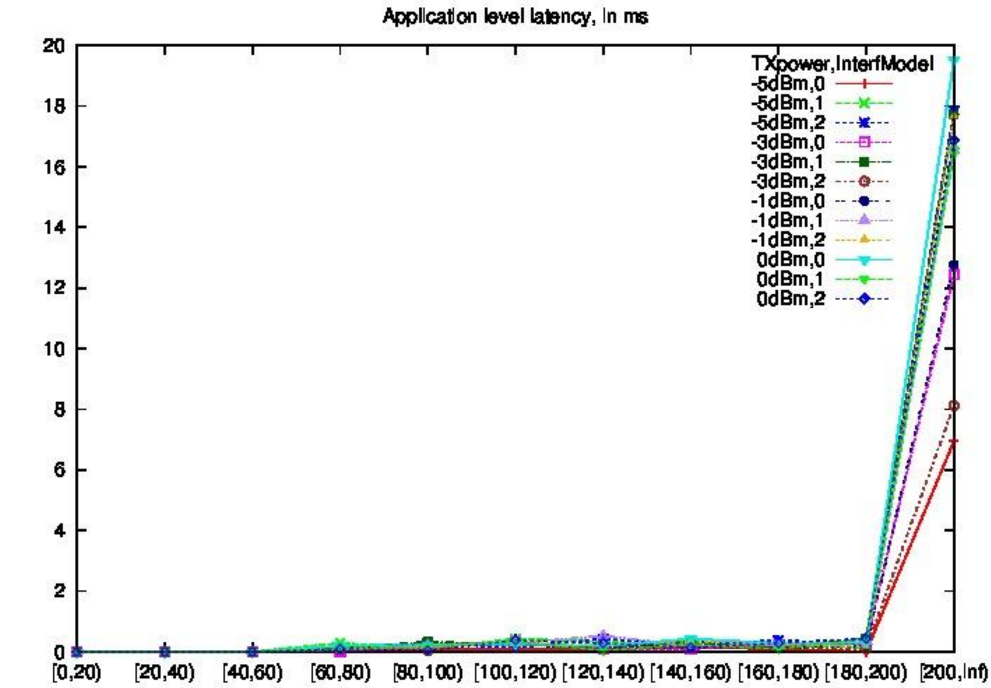

Fig 25: Application Level Latency

LABILE Performance Analysis:

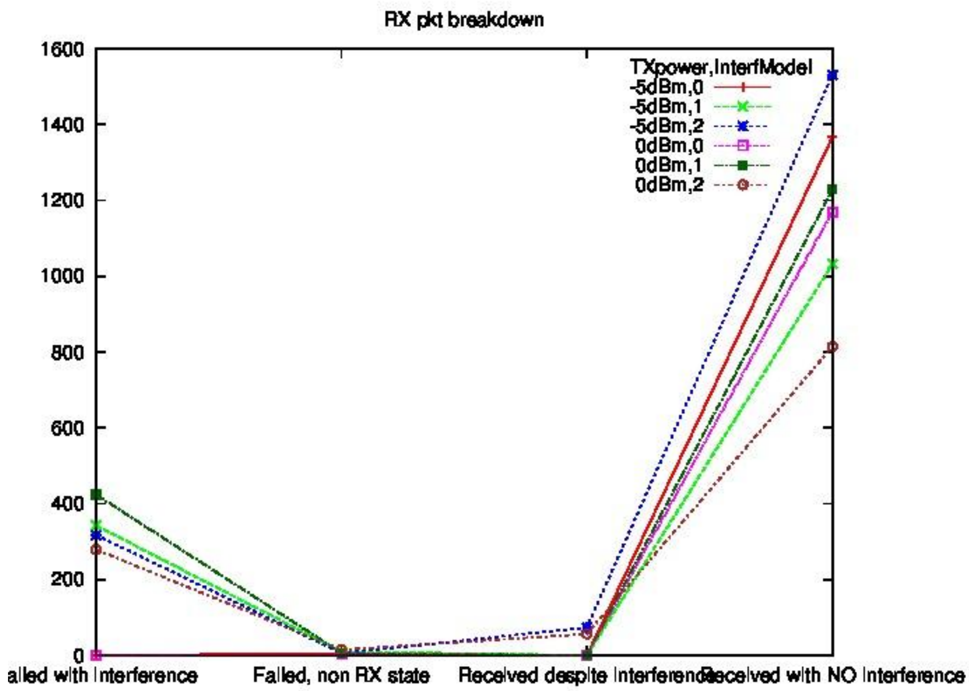

Fig 26: Received Packet Breakdown

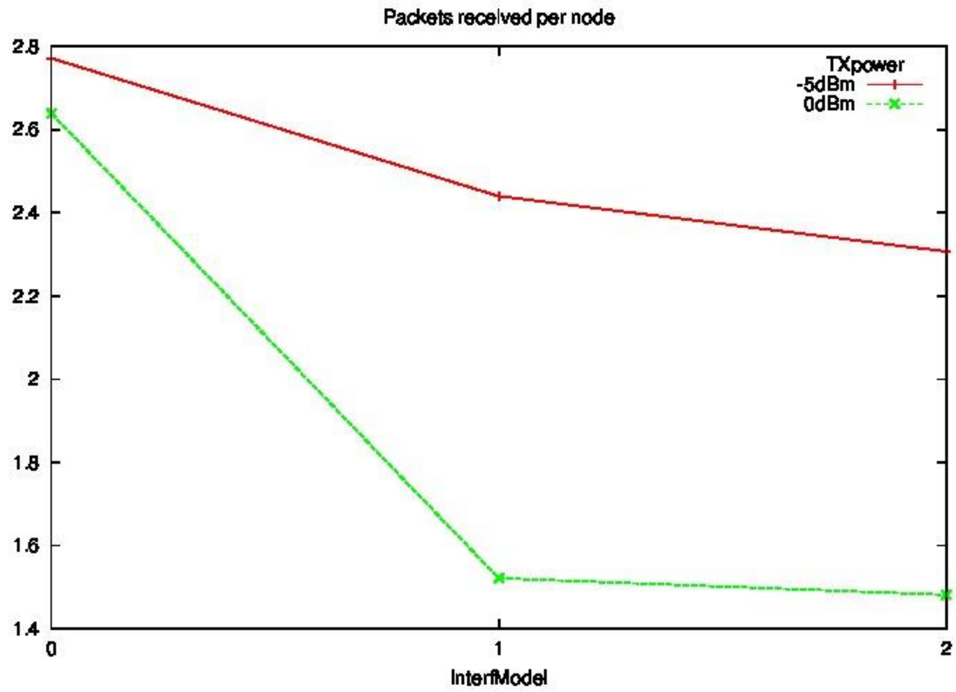

Fig 27: Packet Received per Node

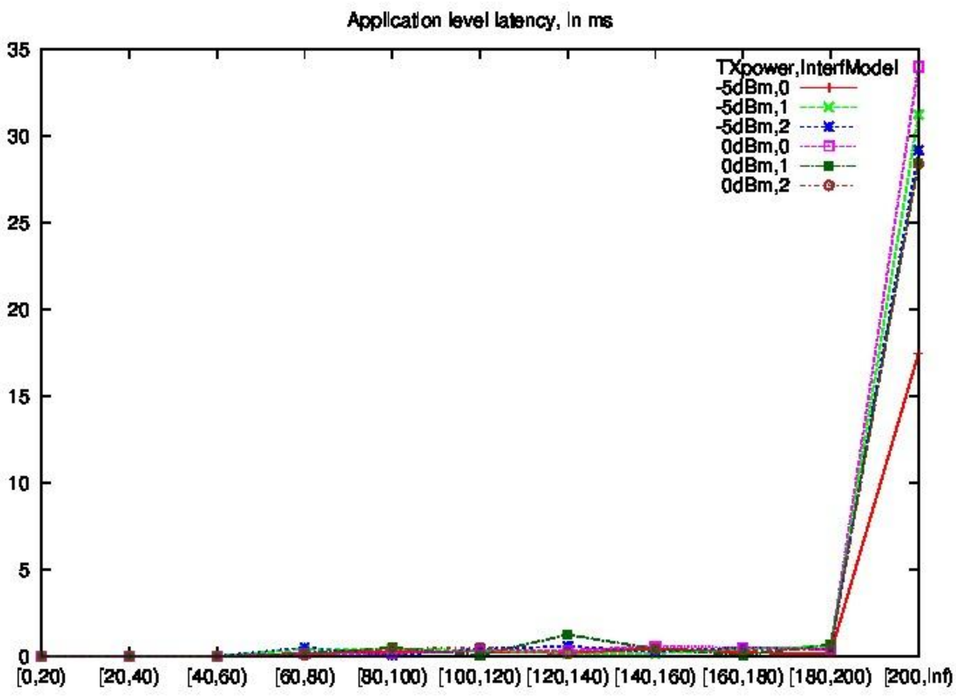

Fig 28: Application Level Latency

REL Performance Analysis:

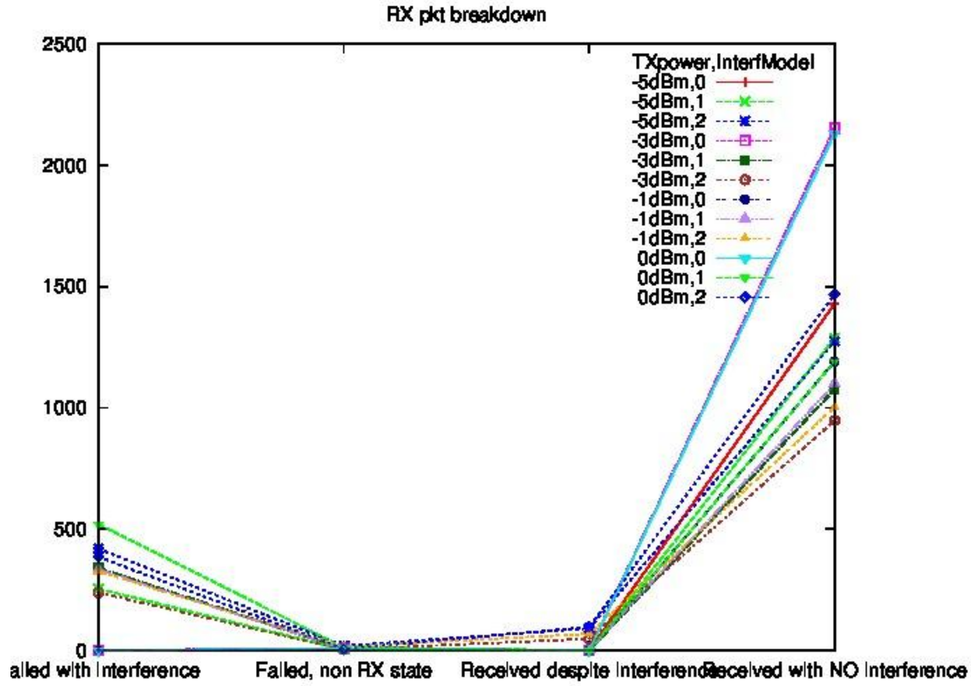

Fig 29: Received Packet Breakdown

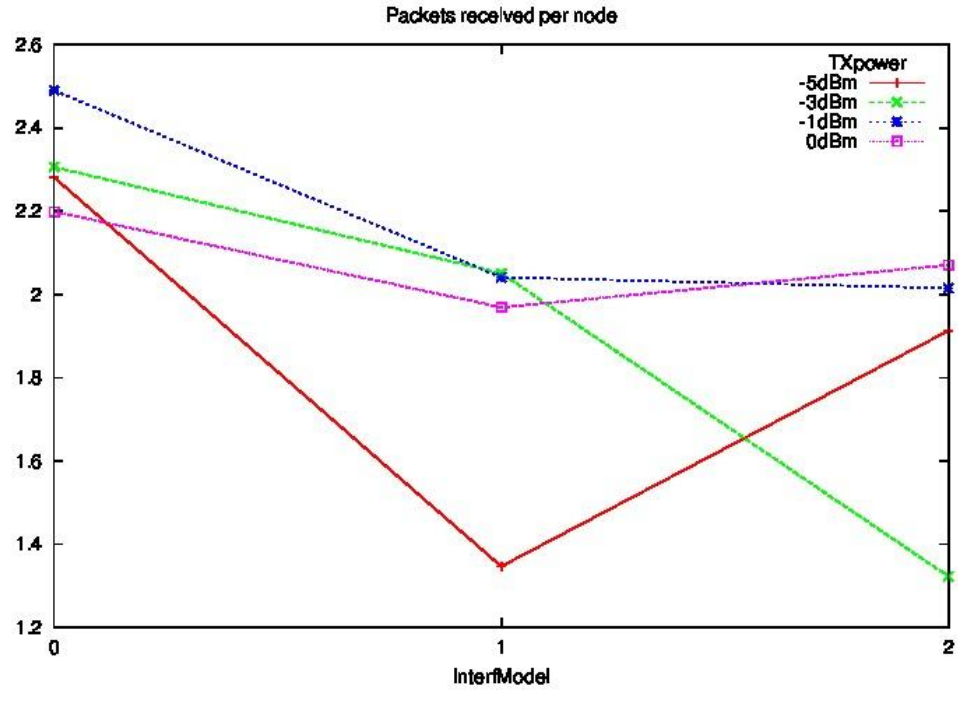

Fig 30: Packet Received per Node

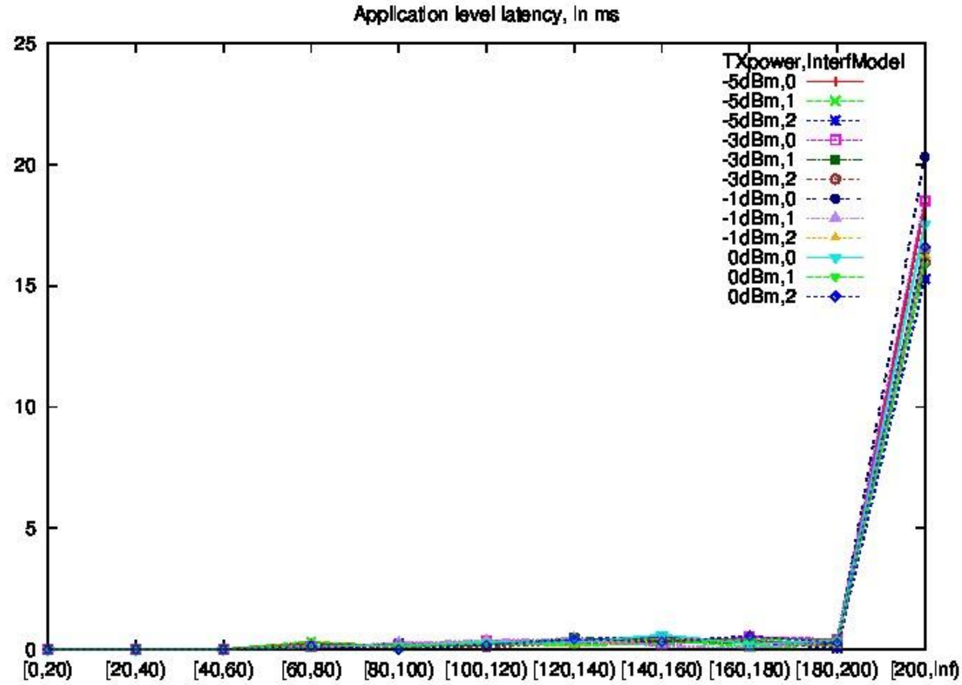

Fig .31: Application Level Latency

OBSERVATIONS

- Here we see that we get a better performance from the wireless sensor network if we use a higher transmission power as seen in simulations run with different interference models.

Routing Protocol Comparison:

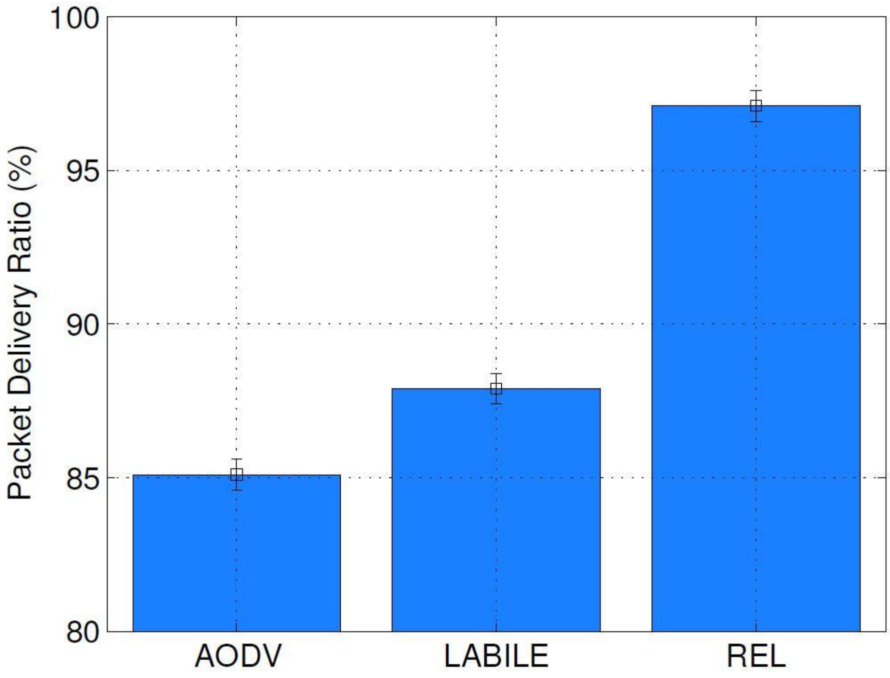

Fig 32: % Packets Delivered

Observations:

Here we can see that REL routing protocol gives better results in our scenario, it has higher no of nodes alive at any time and no of packets delivered is also greater than LABILE and AODV protocol. On the basis of performance analysis of individual protocols we can say that for underwater scenario higher transmission power is required for successful transmission of data.

CONCLUSION

The proposition of a system for water monitoring by using underwater wireless sensor networks is presented. The work is divided into two parts; firstly design of UWSN through by implementing various and comparing multiple routing protocols using specific metrics such as energy consumption, total drop packets, and Packet delivery ratio then elected the appropriate one based on system requirements. Second we implement cloud scenario with the help of CloudSim and compare various allocation policies which could be used with UWSN. We consider various factors which could affect the UWSN. For UWSN we find that REL protocol gives us best results when compared with AODV and LABILE. As for cloud scenario we compare time shared and space-shared policy along with DVFS implementation. And we can conclude that time shared policy is better on the basis of performance evaluation.

REFERENCES

I]Zuniga, Marco, and Bhaskar Krishnamachari. "Integrating future large-scale wireless sensor networks with the internet." USC Computer Science, Tech. Rep (2003).

II] Villegas, Miguel Angel Erazo, Seok Yee Tang, and Yi Qian. "Wireless Sensor Network Communication Architecture for Wide-Area Large Scale Soil Moisture Estimation and Wetlands Monitoring." University of Puerto Rico at Mayaguez WALSAIP RESEARCH PROJECT.

III] The Art of Wireless Sensor Networks: Volume 2: Advanced Topics and Applications. Habib M. Ammari

IV] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience 41, no. 1 (2011): 23-50.

V] Chung, Wen-Yaw, Pei-Shan Yu, and Chao-Jen Huang. "Cloud computing system based on wireless sensor network." In Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on, pp. 877-880. IEEE, 2013.

VI] Guobao Xu, Weiming Shen, and Xianbin Wang, “Applications of Wireless Sensor Networks in Marine Environment Monitoring: A Survey”, Journal- IEEE SENSORS JOURNAL · NOVEMBER 2014