

Hop-by-Hop Dynamic Addressing Based Routing Protocol for Monitoring of long range Underwater Pipeline

Muhammad Zahid Abbas^{1,2,*}, Kamalrulnizam Abu Bakar¹, Muhammad Ayaz³,
Mohammad Hafiz Mohamed¹, Moeenuddin Tariq¹

¹Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

²COMSATS Institute of Information Technology, Vehari, Pakistan

[e-mail: zahidmcs@gmail.com]^{1,2}

[e-mail: knizam@utm.my]¹

[e-mail: mhafiz.mhd@gmail.com]¹

[e-mail: moeenuddinali@gmail.com]¹

³Sensor Networks and Cellular Systems (SNCS) Research Centre, University of Tabuk, Tabuk, KSA

[e-mail: ayazsharif@ut.edu.sa]³

*Corresponding author: Muhammad Zahid Abbas

*Received July 3, 2016; revised October 19, 2016; revised November 22, 2016; accepted December 13, 2016;
published February 28, 2017*

Abstract

In Underwater Linear Sensor Networks (UW-LSN) routing process, nodes without proper address make it difficult to determine relative sensor details specially the position of the node. In addition, it effects to determine the exact leakage position with minimized delay for long range underwater pipeline monitoring. Several studies have been made to overcome the mentioned issues. However, little attention has been given to minimize communication delay using dynamic addressing schemes. This paper presents the novel solution called Hop-by-Hop Dynamic Addressing based Routing Protocol for Pipeline Monitoring (H2-DARP-PM) to deal with nodes addressing and communication delay. H2-DARP-PM assigns a dynamic hop address to every participating node in an efficient manner. Dynamic addressing mechanism employed by H2-DARP-PM differentiates the heterogeneous types of sensor nodes thereby helping to control the traffic flows between the nodes. The proposed dynamic addressing mechanism provides support in the selection of an appropriate next hop neighbour. Simulation results and analytical model illustrate that H2-DARP-PM addressing support distribution of topology into different ranges of heterogeneous sensors and sinks to mitigate the higher delay issue. One of the distinguishing characteristics of H2-DARP-PM has the capability to operate with a fewer number of sensor nodes deployed for long-range underwater pipeline monitoring.

Keywords: Routing Protocol, Hop-by-Hop Routing, Dynamic Addressing, Underwater Linear Sensor Networks (UW-LSNs), Pipeline Monitoring

1. Introduction

Underwater Wireless Linear Sensor Networks (UW-LSNs) provide support for the discovery of several types of underwater applications including underwater pipelines inspection, exploration of aquatic environments and monitoring of underwater boarders [1]. The hazardous aquatic creatures, high water pressure, darkness, low-frequency acoustic communication make the underwater environments harsh to the extent that it remains difficult for human to operate in such environments [2, 3]. In this state of affair, the usage of wireless technology has emerged as the inspiring solution for the exploration of the underwater environment. Linear Sensor Networks (LSNs) and terrestrial WSNs have many shared properties like deployment of wireless nodes and energy constraints, however, UW-LSNs exclusively have some distinctive features as well. Firstly, sensors are without a dynamic address that maintains linear network topology and divide the pipeline length into sub-lengths and ranges of heterogeneous sensors. Secondly, most of the time sensors are deployed randomly and distributed into layers that are not reliable and scalable solution in order to monitor long-range underwater pipelines [4]. Thirdly, higher latency stays common due to the fact that higher frequency waves cannot travel in the underwater environment while acoustic communication remains the only suitable way of communication. Due to the selection of acoustic communication, the signal propagation speed is decreased up to the speed of sound.

Although, the sound waves travel longer and faster in underwater than in air but still they are five times slower than radio waves. Fourthly, proper packet forwarding mechanism is required for the UW-LSN as underwater nodes being larger in number consume more power for data transmission between the sensors and sinks. Lastly, due to low availability of bandwidth, higher workload, and lower data rates stand major problems for such type of underwater networks. Most of the routing protocols are not suitable for the underwater environment as they increase traffic load and delay due to heavily broadcasting during nodes discovery process. Significant areas of improvement are highly required to be there in UW-LSN routing techniques like scalable nodes deployment, dynamic addressing of nodes, minimizing the communication delay, reliable forwarder node selection and efficient data deliveries to the sink, where available acoustic signals data rate is very small and is not practicable for real-time communication. In short, it is hard to increase the data rate for long range underwater pipeline monitoring process except for the usage of distributed topology network. For obtaining more accurate results, it sounds better to utilize the localized nodes with proper addresses rather than using robots or vehicles as it may not be able to bring the real-time results about events in the underwater environment [5].

The Hop by Hop Dynamic Addressing based Routing Protocol for Pipeline Monitoring (H2-DARP-PM) deal with problems mentioned above and provide suitable solutions. The rest of the paper is structured into following sections. Section 2 and 3 presents related work and proposed types of nodes and contributions. Section 4 consists of details about H2-DARP-PM routing protocol design. Section 5 includes an analytical model for deployment of heterogeneous sensors. Performance evaluation of the H2-DARP-PM protocol is provided in section 6. Further, section 7 highlights the important aspects of the H2-DARP-PM routing protocol and section 8 concludes the proposed work providing future directions.

2. Related Work

It is a challenging task to develop a dynamic addressing scheme for scalable nodes deployment which could easily distinguish heterogeneous types of sensors among each other. Further, it is difficult to find the next hop neighbour in order to establish the shortest route towards sink(s) same time maintaining the secondary route in an underwater environment when facing high energy constraint, common topology changes and nodes failure. For such environment, many routing schemes have been proposed to monitor the underwater pipelines while among most of them require special setups and hardware tools like robots or vehicles [6-8]. These routing protocols are classified into two categories; firstly, some protocols require special network setups, extra automotive tools like [7-11] and deploy homogenous types of sensors. All these protocols are in need of an extraordinary system and multiple types of robots/AUVs equipped with a special sensor that moves over the pipeline to collect the data from pipeline sensors deployed at pipeline surface. However, these protocols do not have the suitability for the long term and large scale underwater pipeline monitoring process. The use of homogeneous sensors and automotive tools also increase the cost and delay in the long range monitoring pipelines. While the second type called chain based routing, where schemes mostly require neighbour nodes discovery table for each sensor of the complete network. For the sake of ease, such protocols assume that all nodes in the network already have the detail of their own location and destination information. The requirements of this type of protocols [12, 13] are not easy to be implemented in underwater as localization still remains open research issue.

Table 1. Applications in each class

Algorithm	Requirement/Limitation(s)
SPAMMS[7]	<ul style="list-style-type: none"> i) Robot moves in the pipeline to collect data from RFID sensors deployed at the inner wall of the pipeline for the detection and repairing of defected pipeline location. ii) SPAMMS is suitable for small-scale monitoring.
PIPENET[14]	<ul style="list-style-type: none"> i) Special fixed nodes having static address are required equipped with acoustic pressure sensors. ii) Sink nodes are deployed only on manholes of the sewerage pipeline, hence difficult to find exact leakage place.
SewerSnort[15]	<ul style="list-style-type: none"> i) All the sensors drift inside the sewerage pipeline and in the specific coverage area. ii) The beacons are required to assign node addresses and increase their respective signal strength. iii) It works only for sewerage pipe where fluid flowing at specific speed but not suitable for underwater pipeline monitoring.
TriopusNet[9]	<ul style="list-style-type: none"> i) It needs specific nodes deployment algorithm to release pool of robots in the pipeline. ii) It is not suitable for long range underwater pipeline monitoring as it requires special tools and setup.
KANTARO[8]	<ul style="list-style-type: none"> i) It doesn't require node address as it consists of a fully automatic robot having intelligently motion control tool in it with a scanner and camera that needs to move inside the pipeline. ii) It is the manual way of inspection where the vehicle moves over the pipeline hence hard to install in an underwater environment.
SCADA [16]	<ul style="list-style-type: none"> i) It strictly follows client and server architecture where all clients are attached to the main terminal. ii) This approach involves more manual work therefore not suitable for UWSN.

SWATS [17]	It requires a cross check on the part of the neighbouring nodes along the trajectory of the fluid to validate the node position. It incorporates SCADA system and assumes the static addressing for nodes and control units.
Distributed topology algorithm[18]	i) It is topology discovery algorithm for LSNs that require an ordered list of the node addresses deployed in the network with their relative geographical positions. ii) It works only for the same type of sensors and does not capable of handling heterogeneous sensors.
AUV based algorithm[11]	i) It requires autonomous underwater vehicles (AUV) inside or outside of the pipeline that coordinates with each other according to the application requirement. ii) It requires the intelligent automotive control system to manage trajectory of the vehicle.
Road Monitoring Algorithm[19]	i) It is based on the addresses and placement of nodes on the way of data transmission that makes this technique more effective and efficient for LSN. ii) It is only feasible for homogeneous sensors so heterogeneous types of nodes can't be deployed.
SRJ Algorithm [20]	i) It needs neighbour discovery tables and dynamic signal strengths to bypass failure nodes. ii) Complex routing tables record redirected route status needs to be kept updated.
Chain based algorithm[13]	i) It requires all the nodes connected by wire or virtually bonded in a chain. ii) Each node generates overhead to maintain the record of all other neighbour nodes.

WSNs technologies being effective in data collection are generally used for on-ground oil and gas pipeline monitoring systems. Several authors proposed and discussed regarding data collection in pipeline monitoring system; where mobile relay nodes are normally used for data collection and packet forwarding. In [21] authors proposed an efficient pipeline inspection and forwarded its status information by using data collection algorithm. This algorithm is based on LSN nodes deployment scheme and data fusion strategy in WSN. For on ground pipelines inspection, such techniques can produce remarkable results in improving the network performance but in the case of underwater pipeline inspection due to a large number of random nodes deployment, this technique will become highly expensive. On the other hand, this algorithm provide better data collection efficiency by using multi-hop, data fusion and energy. There is another study [22] that develops a WSN based methodology in order to deploy above ground relay nodes for underground pipeline monitoring. This methodology provides the best placement of relay nodes considering wireless channels, energy consumption, pipeline coverage and transmission levels. This WSN consists of different types of nodes including mobile sensor node, base station, and multiple relay nodes. The relay nodes are deployed on ground at different feasible sites that may work at different heights of receiving antennas. Data collection occurs as the sensor nodes move in the pipe and pass their information towards relay node and finally, relay nodes send data to the base station. As relay nodes are mobile so it is required to keep on sending regular beacons to announce their position to the sensor nodes. For the pipeline coverage and transmission power management, this study also optimizes the relay node. Moreover, this study provides a vision for the development of other types of pipeline monitoring using sensor networks.

In most of the mobility-based relay nodes data collection approaches, the sensing nodes are static at a fixed location and they determine the introduction of relay nodes into the network. In typical monitoring network, data packets are transmitted from slave (end) nodes towards master nodes. Thus, the sensor nodes which are closer to the master node are more likely to relay more packets than other nodes. In order to tackle this issue, master and slave nodes are

designed as mobile devices. In some cases, it is infeasible or even impossible for mobilizing master or slave nodes to deploy mobile nodes like considering underground pipeline monitoring applications; mostly sensor nodes are buried in soil. The data collection about leaks has a tendency to become more complex when there are several mobile nodes in the network. Consequently, it affects the accuracy of leakage position. Therefore, the mobility-based approaches are considered not applicable for underground pipeline monitoring [23]. There is another research considering WSNs in pipeline monitoring application at a smaller scale. It consists of a protocol stack for providing energy efficiency by incorporating of a sleep-wake pattern which enables each mobile node in the network to wake up for short time interval [24]. Most of the times, relay nodes are responsible for the data collection from the nearest sensor nodes. After data collected from sensor nodes, relay nodes accomplish several functions such as data compression, fusion, transmission etc. Normally relay nodes hold a fixed amount of buffer memory used to hold the collected sensing information until it is transferred to the autonomous vehicle or robot as they reach within its transmission range. In this study [25] they do not use a multi-hop approach in order to communicate their collected data to the Network Operation Center(NOC). Relay nodes play an important role in data collection; however, they have limited resources and communication capabilities. Studies show that WSNs can play an important role in underground water pipeline monitoring applications. In this regard, a study [26] consists of fix and mobile sensor nodes as relay nodes and a base station. It highlights the advantages of heterogeneous WSN that can cover a maximum range as compared to the low power relay nodes. The main task of this study is to improve packet delivery and data collection by using network topology. The future direction is also proposed to enhance the pipeline inspection process for emergency and error prone locations.

In comparative analysis perspective, a short summary of some pipeline monitoring techniques and their data collection process through different kinds of routing protocols are described in **Table 1**. The pipeline monitoring is highly important as there is a broad network of pipelines carrying oil and gas which play an integral role in the energy management and economy of many countries. Such as Nigeria has around 5,000 kilometers oil pipelines consisting of more than 4,000 km of different products carrying pipelines while the remaining length belongs to crude-oil pipeline [27]. The average depth of oceans lays around 2.5km to 3km and pipeline length is more than 100 km. In the underwater environment, acoustic communication is considered an ideal solution for communication between underwater sensor nodes. However, if we divide the pipeline length into sub-zones of 1000 meter each, the less number of nodes are required to deliver the data packets from the middle of the pipeline and from bottom to the surface at different ocean depths [28]. It is also important to be noted that the performance of our routing protocol depends on the number and types of sensors. The proposed routing protocol support easily heterogeneous types of sensors but if we increase the number of sensors it will increase the cost of the network. If we use homogeneous types of sensors in that case the acoustic communication will give support up to the range of few km, but it is not desirable as long distance communications utilize more energy. In order to cover the maximum length of the pipeline with low energy and more network lifetime, we have defined the different ranges of acoustic heterogeneous sensors varying from 200 meters to more than 1 km. It is found that acoustic communication work best for the short range applications because it supports a bandwidth of 20-50 KHz [29-32] in 1 km distance. Although, in special cases, the range of sensors can be increased [2], but in normal acoustic communication it is preferred to use the discussed ranges of sensors.

3. Proposed Types of Nodes and Contributions of H2-DARP-PM

This is a well-known fact that most of the linear structures [33] such as pipelines, roads or borders are normally too long; so due to that fact, distributed LSN topology could be scalable and much suitable. The findings from this study reveal that most of the available routing techniques do not distribute network topology and consider whole linear structure (pipeline) as one unit. The concept of LSNs is highly important in linear structures monitoring applications, especially for pipelines monitoring LSN is much appropriate. LSNs can considerably increase the communication efficiency, reliability, fault tolerance, energy savings and network lifetime. Moreover, LSNs are classified according to topological and hierarchical points of view [34]. In topological distribution LSNs are further classified into thin, thick and very thick types of nodes deployment models [1]. The importance of these models varies according to LSN deployment environment and application like in underwater environment, sensors are expensive so in this regard thin deployment model is more suitable.

Although some interesting hop by hop based routing techniques [20, 35, 36] exists in literature but it is cumbersome to implement them for UW-LSNs due to technical and environment limitations. Most of them are designed for homogeneous networks that frequently produce higher delay. In contrast, others intended for heterogeneous networks uses static addressing that doesn't stay much supportive in the routing process. In order to solve the mentioned issues, we propose a novel routing protocol called H2-DARP-PM for long range underwater pipeline monitoring LSNs. The H2-DARP-PM protocol remains helpful to deploy thin LSN model, use heterogeneous types of sensors, assign dynamic addresses, increase packet delivery ratio, and minimize the delay. Four types of nodes are deployed in the proposed network named as Basic Sensing Node (BSN), Data Relay Node (DRN), Data Definition Node (DDN) and Courier Node (CN). BSN has a minimum communication range comparatively while CN has the maximum. All types of nodes are anchored at allotted positions of the pipeline surface while CN is often introduced into the network by using the hydraulic tool. The courier nodes remain helpful in utilizing the better network resources, increasing reliability and minimizing the delay. The courier nodes are also considered as the secondary sinks contribute in providing a secondary path, distributing the network topology, utilizing the better network resources and minimizing the delay. In H2-DARP-PM, courier node is deployed as a surface buoy secondary sink at the top of mid position of the pipeline equipped with both radio and acoustic modems in order to communicate with NOC and heterogeneous types of pipeline sensors. These heterogeneous sensor nodes are deployed according to the LSN topology and placed at the position given by the deployment algorithm[37]. Such nodes are attached with pipeline surface so that they couldn't move freely with the water current while courier node move vertically by using special hardware such as in [35]. All nodes are deployed in a straight line to monitor the pipeline status but their data communication follows the hierarchical model that define the broadcast domains of different types of sensors [18].

In most of the underwater oil and gas pipeline monitoring applications, data packets generation start from the pipeline nodes anchored at the seabed as events often take place around the pipeline[38]. Normally, pipelines are longer in lengths so they face a higher delay to collect the data from farthest nodes to the NOC[39]. In this regard, the usage of heterogeneous and courier nodes are considered favorable. The BSN sensors do the sensing job to inspect the pipeline status while other types of nodes play a role in data dissemination and packet forwarding from BSN towards the NOC. The CN node is introduced in a network for a stated amount of time and after that, it is pulled up towards the ocean surface. The

mechanical tool having positive and negative buoyancy controlled piston performs this task easily. The introduction of courier nodes and dynamic addressing mark the H2-DARP-PM as a state of the art routing protocol because CN being flexible can be introduced at any place of the pipeline in any depth of the ocean. Without courier nodes, it is a challenging task to monitor some areas of the pipelines in the deep and dark sea. In the light of proposed routing protocol, we assume that courier node may reach any area of the pipeline and come back on the surface to any area in the harsh underwater environment. According to the underwater environment conditions and underwater pipeline requirements, the main contributions of H2-DARP-PM routing protocol are as follows.

- I. **Development of dynamic addressing scheme:** Each node obtains its address dynamically and each address consists of specific portions to maintain the heterogeneity of each sensor type, sink address and neighbours distance.
- II. **Efficient packet forwarding process:** H2-DARP-PM provides efficient packet forwarding process by using different types of sensors with different ranges even bypassing the faulty nodes.
- III. **Minimization of the Delay:** H2-DARP-PM divides network topology into sinks and it always forward packets to the nearest sink using less number of hops which ultimately helps to control traffic and minimize the delay.
- IV. **Scalability:** Being scalable, H2-DARP-PM stays extremely supportive for different network sizes or pipeline lengths. Besides, it has the capability to accommodate new nodes, bypass damaged nodes and replace the network nodes without making any serious effect on the rest of network.

4. H2-DARP-PM Routing Protocol

After a broad literature review, it is explored that delay aware UWSN routing is a challenging task [40, 41]. Considering the challenge to minimize the delay, we have proposed H2-DARP-PM that completes its task in two phases. In the first phase, all sensor nodes are assigned the dynamic addresses providing different routes for communication between sensors and sinks. While, in the second phase, data packets are generated and forwarded by BSNs towards the relay nodes and primary or secondary sink by using hop based addresses.

4.1 Addressing Scheme

In H2-DARP-PM, *HopID* (for routing decision) and *Node-ID* (node identifier to maintain linear topology) are used separately. Every node receives and updates its *HopID* dynamically by exchanging hello packets. The *Node-ID* is a distinctive address for each node and sink that remain the same throughout the lifetime of the network. Every *sink* will have two types of addresses:

- **Sink-ID:** a unique *Sink-ID* 10.00.00.00 or 01.00.00.00 for each sink, hello packets use this ID to recognize the sink.
- **HopID:** a static *HopID* "00.00.00.00" is the same for all the sinks. All the data packets use this ID as destination ID.

Sensor nodes of heterogeneous types BSN, DRN, DDN and CN use two types of addresses.

- **Node-ID:** It is a unique *Node-ID* for all the nodes. It helps to recognize them during neighbour node selection and data packets forwarding.
- **HopID:**
 - Ordinary nodes manage and update two types of *HopIDs*, Sink HopID (S-HopID) and Courier HopID (C-HopID). *S-HopID* reaches a maximum of “00.44.44.88”, and each node has this highest value as the default *S-HopID*, before receiving the hello packets. After receiving the hello packet from any sink, it updates new *S-HopID* according to its current location (*section 4.3*). Similarly; *C-HopID* manages the addresses, built with the help of hello packets received from the Courier nodes.
 - Courier node uses a static *HopID* “11.00.00.00”, and CN broadcast their appearance with closet nodes so that they could set secondary path. Courier nodes receive data packets only and do not entertain hello packets from any other node or sink.

4.2 Hello Packet Format

H2-DARP-PM broadcast two types of hello packets, from both sinks (S-hp) and courier node (C-hp).

Sink Hello Packet (S-hp): *S-hp* contains four fields, *Hello Packet Type(s-hp/c-hp)*, *S-HopID*, *Sink-ID(s)* and *Maximum Hop Count* as shown in **Fig. 1**. *Hello Packet Type* field helps to distinguish the type of hello packet that could sink hello packet or courier node hello packet. *S-HopID* consists of eight digits that show how many hops are away from any of the one sink or courier node and also give detail of hop distance of neighbor nodes. *S-HopID* has eight digits like 00.00.00.00 consisting of four groups of two digits in which the first two digits show sink hop distance, the second two digits show DDN hop distance, the third two digits show DRN hop distance and the last two digits show BSN hop distance. In these two digits, left digits have more priority as it establishes a primary route with sink node while right digits are used to establish a secondary route with courier node. *Sink-ID(s)* are used when both sinks broadcast. Hello packets also help to differentiate the both sinks and control duplication of packets received from the same sink. The *Maximum Hop Count* field has an initial value of 00.44.00.00 when sink broadcasts hello packet. After receiving a hello packet, every node updates its own hop-id so that when the hello packet reaches the max hop count value then further broadcasting of this packet is stopped.

Hello Packet Type	S-HopID	Sink-ID(s)	Max. Hop Count
-------------------	---------	------------	----------------

Fig. 1. Sinks hello packet (*S-hp*) format

Courier Node Hello Packet (C-hp): Courier nodes hello packets have five fields as shown in **Fig. 2**. The first field provides details about the hello packet type as discussed under the sink hello packet heading. The *C-HopID* also consists of 8 digits address as S-HopID; these digits show that how many hops the ordinary nodes lay far from the courier node. Courier node broadcasts its ID just like Sinks broadcast their ID in the sink hello packet. Courier node is not fixed on the pipeline utilizing some external tool. It is introduced in the network for only a short period of time so “*Expiry Time*” field is used giving detail about the lifetime of Courier node. All other nodes update their right digits (secondary route) and record the *C-HopID* from hello packets. If they find the courier node closer then deliver the data packets before the expiry time. *Maximum Hop Count* field has the same purpose as in sink

hello packet; it stops the hello packet broadcasting when it reaches the maximum hopes.

Hello Packet Type	C-HopID	Courier-ID	Expiry Time	Max. Hop Count
-------------------	---------	------------	-------------	----------------

Fig. 2. Courier node hello packet (C-hp) format

4.3 Calculating and Assigning the HopIDs

Every ordinary sensor node uses a default value of “11.44.44.88” as its *HopID* and “10.00.00.00 or 01.00.00.00” as its Sink-ID in the routing table until it receives the hello packet. Both the sinks start broadcasting hello packet simultaneously. After receiving this hello packet, all the other types of nodes start updating their own *HopID*. If any node receives hello packet directly from the sink then it updates its *S-HopID* left digit as 10.00.00.00. It then shows that the position of the node as one hop away from the primary sink and more hops away from any other sink. After updating its own address, each node also updates *S-hp* by adding its new *S-HopID*. This process repeats with all neighbour nodes; they update *S-HopIDs* and forward the updated hello packet to their neighbours. The Hello packet broadcast stops only when it reaches the maximum hop count. In our proposed topology, we used two sinks on both ends of the pipeline and their broadcast domain does not conflict with each other as the left sink covers till mid of the pipeline; the rest of the pipeline is covered by the right sink. These both sinks have their unique *Sink-ID*. The whole process is illustrated in Fig. 4 and further clarified in Algorithm 1 for assigning these *HopIDs*. The example of a node address distribution is shown in Fig. 3.

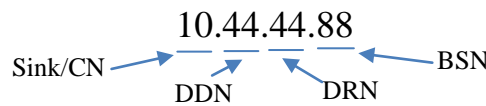


Fig. 3. A node address distribution

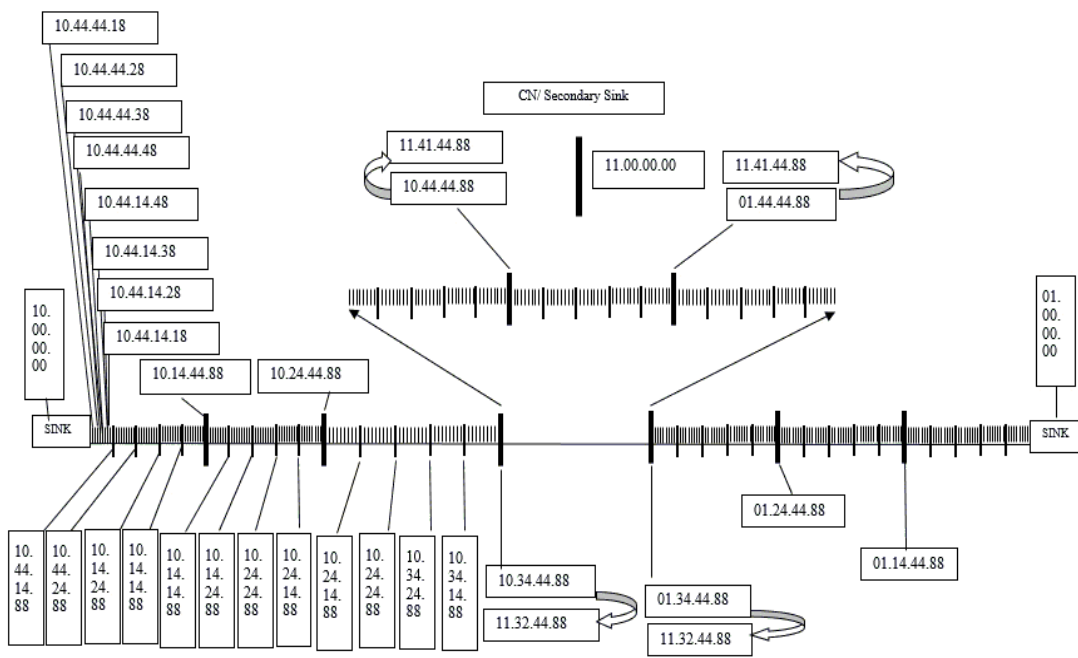


Fig. 4. Assigning HopIDs with the help of hello packet

Algorithm 1. Algorithm for assigning the *HopIDs* with the help of hello packets

Two types of Hello packets (*hp*) broadcasts sink hello packet(*S-hp*) and courier hello packet(*C-hp*)

From each Sink (**S-hp**) with HopID “ $N_{10.00.00.00}$ ” or “ $N_{01.00.00.00}$ ”
 Max Hop Count = 8 for BSN
 Max Hop Count = 4 for DRN
 Max Hop Count = 4 for DDN

From Courier node (**C-hp**) with HopID “ $N_{11.00.00.00}$ ” &
 Max Hop Count = 8 for BSN
 Max Hop Count = 4 for DRN
 Max Hop Count = 4 for DDN

Default nodes addresses:

Right SINK ----- 01.00.00.00
 Left SINK ----- 10.00.00.00
 Courier Node ----- 11.00.00.00 (if more than 1, then increase like 12, 13... etc)
 DDN node ----- 11.44.44.88
 DRN node ----- 11.44.44.88
 BSN node ----- 11.44.44.88

//Hello packet received from Left and Right SINK,s for primary path setting and after completion of this process, Courier node/CN nodes broadcast *hp* for secondary path setting

Notations

- 1) *S-hp-S-leftdigit*: Left digit of sink portion in *S-hp* HopID
- 2) *S-hp-S-rightdigit*: Right digit of sink portion in *S-hp* HopID
- 3) *S-hp-DDN-leftdigit*: Left digit of DDN portion in *S-hp* HopID
- 4) *S-hp-DDN-rightdigit*: Right digit of DDN portion in *S-hp* HopID
- 5) *S-hp-DRN-leftdigit*: Left digit of DRN portion in *S-hp* HopID
- 6) *S-hp-DRN-rightdigit*: Right digit of DRN portion in *S-hp* HopID
- 7) *S-hp-BSN-leftdigit*: Left digit of BSN portion in *S-hp* HopID
- 8) *S-hp-BSN-rightdigit*: Right digit of sink portion in *S-hp* HopID
- 9) *Node-ID-S-leftdigit*: Left digit of sink portion in node own HopID
- 10) *Node-ID-S-rightdigit*: Right digit of sink portion in node own HopID
- 11) *Node-ID-DDN-leftdigit*: Left digit of DDN portion in node own HopID
- 12) *Node-ID-DDN-rightdigit*: Right digit of DDN portion in node own HopID
- 13) *Node-ID-DRN-leftdigit*: Left digit of DRN portion in node own HopID
- 14) *Node-ID-DRN-rightdigit*: Right digit of DRN portion in node own HopID
- 15) *Node-ID-BSN-leftdigit*: Left digit of BSN portion in node own HopID
- 16) *Node-ID-BSN-rightdigit*: Right digit of sink portion in node own HopID

Input

Get received *S-HopID* “ $N_{ab.cd.ef.gh}$ ” from *S-hp*

$a = S\text{-hp-S-leftdigit}$ $b = S\text{-hp-S-rightdigit}$ $c = S\text{-hp-DDN-leftdigit}$ $d = S\text{-hp-DDN-rightdigit}$
 $e = S\text{-hp-DRN-leftdigit}$ $f = S\text{-hp-DRN-rightdigit}$ $g = S\text{-hp-BSN-leftdigit}$ $h = S\text{-hp-BSN-rightdigit}$

Node own HopID “ $N_{wz.tu.rs.pq}$ ”

w = Node-ID-S-leftdigit z = Node-ID-S-rightdigit t = Node-ID-DDN-leftdigit
u = Node-ID-DDN-rightdigit r = Node-ID-DRN-leftdigit s = Node-ID-DRN-rightdigit
p = Node-ID-BSN-leftdigit q = Node-ID-BSN-rightdigit

Process

1. **If** Packet Type = **S-hp**
- Switch (packet type) CASE 1: SINK-DDN Addressing**
2. **procedure** SINK-DRN ADDRESSING(hello packet Hop-ID, node Hop- ID)
3. **for each** DDN node
4. **do**
5. **if** (hp Sender == hp-SINK) && (receiver node own_id == def_id) //Primary path setting
6. { **if** a=0 && b=1 || a=1 && b=0) && (a=w && b=z || a=w && b=z || a=w && b=z) // check hp that is it from SINK but not from the same SINK?
7. w=a //update SINK address 1st bit in own SINK hop address bit
8. z=b //update SINK address 2nd bit in own SINK hop address bit
9. **if** (c<t && t=def) // if hp receiver in DDN and its 1st bit is smaller than current bit
10. t=c // replace current bit with received hp value
11. c++ //increment in hop count value of hp address
12. Repeat steps 9-11 when s-hp received by DRN and BSN compare their 1st bit with S-hp hop id if its smaller then replace its value with s-hp hop value.
13. **as reach** max hop count } //stop the iteration
14. **else{ if** (a=1 && b=1) // Secondary/Secondary Path Setting when CN broadcast hp
15. {w=a //update SINK/CN address 1st bit in own SINK address bit
16. z=b //update SINK/CN address 2nd bit in own SINK address bit
17. **If** (d<u && d=def) // if hp receiving DDN address 2nd bit is smaller than current bit
18. u = d // replace current bit with received hp value
19. d++ //increment in hop count value of hp address
20. Repeat steps 17-19 when hp received by DRN and BSN compare their 2nd bit with C-hp hop id if its smaller then replace its value with C-hp hop value.}
21. **else** {max hop count } //stop the iteration
22. **end for**
23. **end procedure**

CASE 2: DDN-DRN Addressing

24. **procedure** DDN-DRN ADDRESSING(hello packet, node ID)
25. **for each** DRN node
26. **do**
27. **if**(hp_type == hp-ddn) && (own_id == def_id)
28. { **if** (a=0 && b=1 || a=1 && b=0) && (a=w && b=z && c=∅ && d=∅) //Primary path setting
29. w=a
30. z=b
31. t=c
32. u=d
33. **if** (r<e)
34. r=e
35. r++
36. Repeat steps 33-35 when hp received by DRN and BSN compare their 1st bit with S-hp hop id of BSN portion if its smaller then replace its value with s-hp hop value.
37. **as reach** max hop count } //stop the iteration
38. **Else** { **if**(a=1 && b=1) // Secondary Path Setting
39. {w=a

```

40. z=b
41. t=c
42. u=d
43. If (s<f)
44. s=f
45. s++
46. Repeat steps 43-45 when hp received by DRN and BSN compare their 2nd bit with C-hp hop id if
    its smaller then replace its value with C-hp hop value.}
47. else{max hop count } //stop the iteration
48. end for
49. end procedure

```

CASE 3: DRN-BSN Addressing

```

50. procedure DRN-BSN ADDRESSING(hello packet, node ID)
51. for each BSN node
52. do
53. if (hp_typ== hp-drn) && (own_id== def_id)
54. { if (a=0 && b=1 || a=1 && b=0) &&(a=1 && b=1 &&c=0 &d=0 &e=0 &f=0) //Primary path
    setting
55. r=e
56. s=f
57. if (p<g)
58. p=g
59. p++
60. Repeat steps 57-59 when hp received by other BSN compare their 1st bit with S-hp hop id of BSN
    portion if its smaller then replace its value with s-hp hop value.
61. as reach max hop count } //stop the iteration
62. Else{ if(a=1 && b=1) //Secondary Path Setting
63. {r=e
64. s=f
65. If (q<h)
66. q= h
67. q++
68. Repeat steps 65-67 when hp received by other BSN compare their 2nd bit with S-hp hop id of BSN
    portion if its smaller then replace its value with s-hp hop value.}
69. else {max hop count }} //stop the iteration
70. end for
71. end procedure
72. end Switch
73. if Packet Type = SINK-hp // Default setting of hp broadcasting and general updating of each
    node address
74. Max. Hop Count - 1
75. if Max Hop Count > 0 then //if maximum hop count is not equal to zero
76. Update SINK-hp ← Own SINK-HopID //each node update its own SINK hopID with
    SINK-hp
77. Broadcast SINK-hp further //broadcast hello packet to other neighbor nodes
78. else
79. No further broadcast for this SINK-hp
80. end if
81. end if
82. if Packet Type = C-hp // similar steps repeated for courier node hello packets
83. Max Hop Count - 1

```

```

84.   if   Max Hop Count > 0 then
      Update C-hp ← Own C-HopID
      Broadcast C-hp further
85.   else
86.   No further broadcast for this C-hp
87.   end If
88.   end If
    
```

Output: All nodes have updated their own Hop IDs after receiving hello packets from sink or courier node.

Likewise, *S-hp* all ordinary nodes receive *C-hp* of the courier node. The process of each node’s hop address updating is in similar manner as described by sink hello packet updating process. The core purpose of courier node is to establish a secondary path in order to minimize the delay whereas it is deployed on top of the middle area of the pipeline because middle area nodes being farthest from both the sinks face a higher delay. *C-hp* updates only the right hop digit of each node address and establishes the secondary route for all the ordinary types of nodes. Each node updates its right digit to set its secondary path and forward the *C-hp* with new *C-HopID*. It is also required that all nodes record their courier node *Node-ID* and then update the *C-HopID*. Now each node has two routes i.e. primary route is established toward primary sink and secondary route towards the courier node. The proposed addressing scheme makes it easy for each node to calculate the shortest route either towards the sink or courier node. This idea works by checking the neighbor nodes hop address left and right digits. This is important to clarify the reason of using the secondary route for a courier node. Firstly, the long length of pipelines increases the delay in underwater communication between nodes and the sinks. Secondly, it does not remain feasible to deploy sink node near the middle of pipeline in the deep sea. Therefore, the introduction of courier node stays much helpful to decrease the delay in data collection from middle nodes of the pipeline. We have used one courier node but in the case of thousand miles pipeline length, more courier nodes can be easily added to this network. We have also tried to minimize the workload on the ordinary nodes as they retain only one entry routing table as shown in **Table 2**.

Table 2. Address update for Primary path setting with sink

Hello Packet Type & hop ID	SINK-DDN Addresses	SINK-DRN	INK-BSN	DDN-DRN	DDN-BSN	DRN-BSN
Default hop address in hp (00.00.00.00)	01.14.00.00	01.00.14.00	01.00.00.18	01.14.14.00	01.14.00.18	01.14.14.18
	01.24.00.00	01.00.24.00	01.00.00.28	01.14.24.00	01.14.00.28	01.14.14.28
	01.34.00.00	01.00.34.00	01.00.00.38	01.14.34.00	01.14.00.38	01.14.14.38
	01.44.00.00	01.00.44.00	01.00.00.48	01.14.44.00	01.14.00.48	01.14.14.48
			01.00.00.58		01.14.00.58	01.14.14.58
			01.00.00.68		01.14.00.68	01.14.14.68
			01.00.00.78		01.14.00.78	01.14.14.78
			01.00.00.88		01.14.00.88	01.14.14.88

Table 2 discusses an idea that how sink hello packet circulate and update addresses of the different types of nodes. Before receiving a hello packet, all nodes consist of default address (11.44.44.88). After arrival of the hello packet, each node starts updating the left bit of hop ID

as shown in Fig. 6. For example, if sink or CN sends a packet to DDN then DDN node updates its first two defaults bits (11) according to hello packet sink ID so it becomes either 01(left SINKID) or 10(right SINKID). After that it updates DDN hop address portion to 14 from 44 (default address). This updated address 14 of DDN portion means that sink lays one hop away on the primary path. Further, it broadcast hello packet to the next hop node with updated hop address; when the next DDN receive hello packet then it updates its address 24 and so on. A similar pattern is followed by DRN and BSN nodes for updating of their hop addresses. All nodes update only left bit of their addresses as a left bit is utilized for primary route establishment either towards the left or right SINK. The hello packet broadcasting process starts from the left and right SINKs simultaneously and stops at the maximum hop count value till mid position of the pipeline. The total length of the pipeline is equally divided into two sinks; left SINK to get coverage from the left corner till the mid of pipeline while the right SINK is covered from right corner till the mid of pipeline.

Table 3. Address update for secondary path setting with courier node

Hello Packet Type & hop ID	CN-DDN Addresses	CN-DRN	CN-BSN	DDN-DRN	DDN-BSN	DRN-BSN
Default hop address in hp (00.00.00.00)	11.14.00.00	11.00.14.00	11.00.00.18	11.14.14.00	01.14.00.18	01.14.14.18
	11.23.00.00	11.00.23.00	11.00.00.27	11.14.23.00	01.14.00.27	01.14.14.27
	11.32.00.00	11.00.32.00	11.00.00.36	11.14.32.00	01.14.00.36	01.14.14.36
	11.41.00.00	11.00.41.00	11.00.00.45	11.14.41.00	01.14.00.45	01.14.14.45
			11.00.00.54		01.14.00.54	01.14.14.54
			11.00.00.63		01.14.00.63	01.14.14.63
			11.00.00.72		01.14.00.72	01.14.14.72
			11.00.00.81		01.14.00.81	01.14.14.81

Due to higher delay and lower bandwidth support in an underwater environment, the middle nodes surely face higher latency problem being farthest from both the Sinks. In this regard, we have introduced a higher frequency courier node installed on the surface by floating buoy on top of the mid of pipeline. Courier node also broadcast hello packet to establish the secondary route by updating the hop address bit of all the nodes. Table 3 presents the address updating process of the C-hp. In this process after receiving the courier hello packet(C-hp), each type of nodes update their right bit of own hop address and establish secondary route toward the courier node.

4.4 Data Packet Format

The data packet format for H2-DARP-PM is presented in Fig. 5. Each data packet consists of four control fields, namely; Source Node-ID, Next hop Node-ID, Packet Sequence Number, Destination ID and one Data field consisting of sensed data.

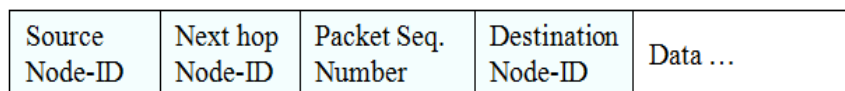
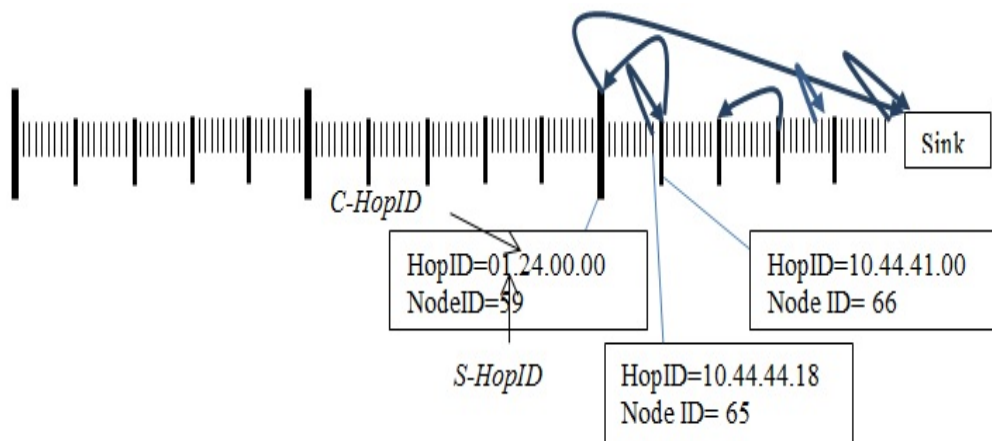


Fig. 5. H2-DARP-PM data packet format

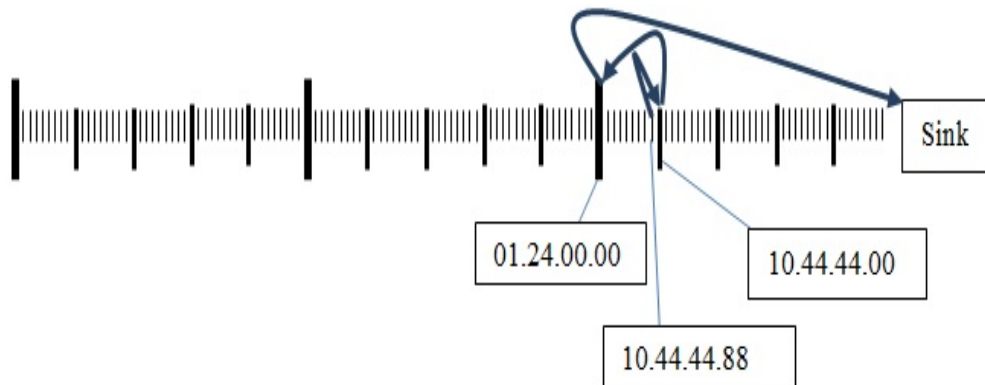
Source Node-ID is used to share the *Node-ID* of that node having the data packet and wants to be transferred it to the sink or courier node. Before forwarding the data packet, each source node collects details of its neighbor nodes and selects *Next hop Node-ID* for packet forwarding. The forwarder node is then selected among the neighbor nodes being nearer to either the sink or the courier node. *Packet Sequence Number* is assigned by a source node for maintaining the integrity of the data and to be on the safe side from packet loss. *Destination ID* contains one value among these three “01, 10, 11”, being the *ID* of either sink or courier node. These IDs therefore, help in delivering the data packets to any one of the closest sink or courier node.

4.5 Data Packet Forwarding Process

Fig. 6(b) explains data packet forwarding process that how source node selects the next hop for the data delivery. A source node N65 has a data packet with its own *HopIDs* 10.44.44.18 (*C-HopIDs* is default=8 which means there is no courier node available in this area); it broadcasts “*Inquiry Request*” message to its neighbours in order to request about their *HopIDs*. This “*Inquiry Request*” message contains only the *Node-ID* of the requesting node. After getting this inquiry message, all neighbours nodes respond as “*Inquiry Reply*” message having four fields, *Node-ID*, *Node Type*, *S-HopID*, *C-HopID* of replying nodes. For example, nodes N63, N64, N66, and N67 are in the range of source node (N65) so they inquiry reply with their *Node-IDs* and *HopIDs*. After receiving all neighbour nodes replies, N65 lists down all the *Inquiry replies* and at first try it attempt to select the higher level neighbour node; if it doesn't find the higher neighbour node then selects the same level of a node closer to the sink/courier node. The **Fig. 6(b)** shows nodes N66 and N67 as declared candidates for the Next Hop because both of these have smaller *S-HopIDs* as compared to *S-HopID* of the source node. In this regard, N66 wins the competition due to the higher level node (DRN). The BSN type of a source node (N65) selects the next hop and forwards the data packet to the DRN type node N66 in the range of source node. In some cases, if all neighbours are of the same type of nodes and possibly have the minimum *HopIDs*, then the node closer to the sink stands as eligible to be selected as the next hop. **Fig. 6(a)** presents almost the similar procedure as shown in **Fig. 6(b)**, only with the addition of following flat communication model; whereas there is no need to follow the hierarchy of nodes and all types of nodes communicate with each other existing at the same level or different levels.



(a) Selecting the Next Hop in primary path toward sink (in flat communication model)



(b) Selecting the Next Hop in primary path towards sink (in hierarchical communication model)

Fig. 6. Selecting the Next Hop node for the data delivery process

In case two BSN source nodes (N65 and N88) are ready to send their data packets simultaneously towards the sink, then N88 requests its neighbours for their *HopIDs*; N86, N87, N89 and sink being in its range, reply with their *HopIDs*. *C-HopID* right digit of its neighbour nodes IDs is still default showing the unavailability of courier node so the source node forward the packet to N89 or direct towards the primary sink. If N89 receive packet, then similar process is repeated for the selection of next hop and it continues till the data packet reaches the sink. In this scenario, courier node is not available or lies at the higher hops distance from these nodes as compared to the sink.

Another scenario is showing in **Fig. 7** that demonstrates the packet forwarding process with the availability of courier node. The *HopIDs* of few nodes is shown in **Table 4**. In this case, N68 being the source node broadcast inquiry request to its neighbor nodes in order to get their details. Here N69, N67 and N70 as its closest neighbors, reply their details to the source node. The availability of the courier node is shown in their replied *HopID*. After this, the source node sorts out the neighbor nodes *HopIDs* list and select the next hop with a minimum number of hops involved to reach any sink or the courier node. **Fig. 10** and neighbor nodes addresses show the courier node closer to the source node than the sink. In this situation, N69 is upper-level DRN node with smaller *C-HopID* considered as the next hop. After receiving the data packet, N69 repeats the same procedure and continue it till the data packet reaches the courier node. The data packet forwarding process is further defined in **Algorithm 2**.

In efficient packet forwarding, it is highly important to be considered the failure nodes. As it is not possible that always source node get response from the neighbour node having smaller *HopIDs*. In this regard, most of the times neighbour nodes do not respond to source node's inquiry request, particularly in sparse nodes deployments. Therefore, H2-DARP-PM efficiently addressed these issues by adding the waiting time in inquiry request; this time is defined in section (**Section 4.6**) where source node tries three times to get the neighbour *HopIDs*. After the third try, if no response receives then the source node consider that no neighbour node of higher level is available. Consequently, it forward the data packet towards the node on the same level shown in **Fig. 6(a)** with the *HopID* closer to sink. These three attempts to an upper-level node are helpful to get the refined results of packet delivery ratios and end-to-end delays. There is also a chance of worst cases when source node is not able to communicate with any other node; this issue is resolved by changing the communication range and source node tries to get inquiry replies of next nodes to its neighbours [3, 42].

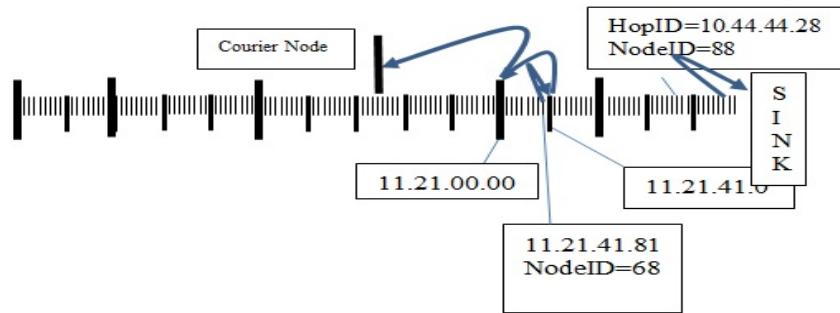


Fig. 7. Selecting the Next Hop in secondary path toward CN

Table 4. HopIDs of nodes

Node-ID	Node Type	-HopID (left bit)	-HopID (right bit)
N66	BSN	10.21.41.61	11.21.41.61
N67	BSN	10.21.41.71	11.21.41.71
N68	BSN	10.21.41.81	11.21.41.81
N69	DRN	10.21.41.00	11.21.41.00
N70	BSN	10.21.41.11	11.21.41.11
N87	BSN	10.44.44.38	10.44.44.38
N88	BSN	10.44.44.28	10.44.44.28
N89	BSN	10.44.44.18	10.44.44.18
N40	BSN	10.31.41.81	11.31.41.81
N41	DRN	10.31.21.00	11.31.21.00
C1	CN	N/A	11.00.00.00
N60	DDN	10.21.00.00	11.21.00.00
N61	BSN	10.21.44.81	11.21.44.81

The packet delivery ratios for H2-DARP-PM do not fully depend on the network density or sparseness. Moreover, we improve its performance by using distributed topology network and heterogeneous types of sensors. To minimize the traffic load, we define that after receiving the Inquiry Request message, only the hierarchically higher level neighbour nodes and same level nodes with smaller HopIDs may reply. In the case of a sparse network, all the neighbour nodes respond to the Inquiry Request.

Algorithm 2. Forwarding the data packets with the help of HopIDs

Input: Data packet (*dp*) ready to send (BSNs initially generate *dp* and forward to next hop then same procedure repeats by each next hops till *dp* arrive at any sink)

Process:

1. **Procedure** PACKET FORWARDING(*dp*, next hopIDs)
2. **If** Next Hop \neq Expire
3. Forward *dp* to Next Hop
4. **If** Ack. Received = Yes
5. **If** next *dp* ready = yes
6. go to step 2
7. **Else**
8. wait until *dp* ready
9. **End If**
10. **Else**

11. go to step 14
 12. **End If**
 13. **Else**
 14. Request Count for this $dp = 0$
 15. Request neighbours for HopIDs
 16. Req. Count+1
 17. **If** Current HopID $\geq dp$ Source HopID & Request Count < 3 **Then***
Discard Inquiry Reply of Source node
 18. Replied HopIDs put in array
Sort out and get the Minimum of both S-HopID and C-HopID
(Min. S-HopID & Min. C-HopID)
 19. **Else**
 20. Replied HopIDs put in array
Sort out and get the Minimum of both S-HopID and C-HopID
(Min. S-HopID & Min. C-HopID)
 21. **End If**
 22. Compare (Min. S-HopID) & (Min. C-HopID) and get Smaller
(SML-HopID)
 23. **If** Request Count < 4 **Then**
 24. **If** SML.HopID $<$ Own HopID **Then**
 25. Next Hop \leftarrow Node-ID of SML.HopID
 26. go to line 2
 27. **Else**
 28. Wait defined amount of time //Section 4.6
 29. go to step 15
 30. **End If**
 31. **Else**
 32. Forward dp to Node-ID of SML.HopID, Until packets in buffer, are available
 33. **End If**
 34. **End Procedure**
- Output:** All generated data packets delivered to either the sink or courier node.

* Is current dp received from an upper-level node? If so then don't consider the inquiry reply from that source node.

4.6 Calculating the Waiting Time

When a source or forwarding node does not find next hop neighbour belonging to the upper-level nodes at first attempt then it may take two more tries to find upper-level nodes before sending it towards the same level nodes. We have assigned waiting time intervals that values are between [0 and 10], where 0 means no wait and 10 is the maximum wait in the worst situations. We calculate the waiting time i.e. $Time_1$ in (1) against the 1st inquiry request depending upon the number of nodes reply. In the case of getting no any response from upper-level neighbour nodes, the 2nd time inquiry request is broadcasted.

$$Time_1 = \frac{K}{n_1 + 1} \quad (1)$$

Here K stays constant that has the maximum value of waiting time between 0-10 and n_1 represent the total nodes replies in the first inquiry request. After the 2nd try, if it still does not find any node from the upper layers then it waits for $Time_2$ as in (2) depending on two

parameters; firstly, the number of nodes replayed after the 2nd inquiry request and secondly, the difference between the number of nodes in the 1st and 2nd inquiry request. We then get the average of these parameters.

$$\text{Time}_2 = \left[\frac{K}{|n_2 - n_1| + 1} + \frac{K}{n_2 + 1} \right] \quad (2)$$

Here K is a constant and n_2 is the number of neighbour nodes replied in the 2nd request. From (1) and (2) it is clear that the waiting time depends on the accessibility of higher level neighbour nodes and a number of the nodes reply. These parameters work inversely to the waiting time because when the waiting time begins to decrease then one or both of them begin to increase.

4.7 Route Updating and Maintenance

Due to the sparse networks and energy issues in underwater environment, nodes either die or fail to communicate ultimately not being able to participate in packet forwarding process. While this happens, the source node increases its range and able to trace the nodes next to its immediate neighbours, both at upper and same level nodes. Each node ID consists of complete details such as the type of the node giving a response and number of hops required to reach the sink or courier node. It is not always necessary to deliver data packets to the primary sink only as data packets can also be delivered to courier node using secondary path.

Although the proposed addressing scheme can be used for any kind of LSN but we focus particularly on the underwater oil and gas pipeline monitoring application, where sampling of the data for pipeline status is mostly required after frequent intervals. In order to conserve energy resources, nodes only operate during data sampling, afterwards nodes switch to sleeping mode or shutdown transceiver. The time intervals for sensing and sleeping are scheduled depending upon the application environment and requirements of the network. The underwater pipeline monitoring networks are mostly designed for the long range and periodic times. These sampling intervals do not require changing of HopIDs, as nodes are static and fixed with the pipeline surface. In the current scenario, we have used one courier node but if the number of courier nodes increase then updating of HopIDs may also be required. For long-term monitoring operations; these HopIDs are updated at the time of introducing new courier nodes in the network. In the case of courier node leaving the network, the HopIDs of the secondary path reset the default values throughout the network; but if again a new courier enters the network then it broadcast new hello packets for updating the secondary path HopIDs of the closer nodes.

We also suggest a strategy for overcoming the failure nodes issue by extending the range of source node bypassing the failure nodes. If source node still does not get any response, then it further increases its range until it reaches the maximal value. If even in maximal range the failure nodes and broken links are not bypassed then in that case, the message is dropped. We have proposed the maximal range of the each type of sensor and is represented as $M=3$ telling that a node can bypass the maximum number of 3 broken links or failure nodes.

5. Performance Evaluation

In this section, we have evaluated performance of the H2-DARP-PM through extensive

simulations using Aqua-Sim (NS-2 based) simulator. For this purpose, the considered simulation parameters are listed in **Table 5**. The process of data delivery is completed by a minimum number of hops with three heterogeneous types of sensors starting from the BSN towards the sink or CN. The deployment of CNs is optional as our proposed routing protocol can complete its task without their presence. However, for better resource usage, we used a small number of CNs i.e. 1% or 2 % of total the sensor nodes. All sensor nodes are static and fixed with the pipeline surface where they target to forward their data horizontally either towards the left or right sink while CN moves above the mid area of the pipeline so that it could collect data vertically from pipeline attached sensors. BSN node generates a data packet; that is collected by DRN nodes and forwards it to the nearest DDN node. Finally, DDN forwards the data packets to the sink or CN but it is not compulsory that all nodes should strictly follow the hierarchy. Hence, any type of node is capable of sending or receiving a data packet from any other type of node.

Table 5. Simulation Parameters

Parameters	Values
MAC	Underwater MAC
Transport layer protocol	UDP
Antenna	Omni-Directional
Propagation Model	Underwater Propagation
Channel Bandwidth	25Khz
Number of nodes	120-360
Network dimension	12600m × 500m
Primary sinks location	At both end of the pipeline
Types of nodes	BSN, DRN, DDN,CN
Ranges of nodes	100,250,400,500 m
Maximum Pipeline length	12600 m
Hello packet size	12 byte
Simulation time	1000 sec

5.1 Performance Metrics

We have used data delivery ratios, end-to-end delays, routing overhead and coverage as the metrics in order to examine the performance of the H2-DARP-PM protocol. Delivery Ratio is the total number of data packets received successfully at all the sink(s). End to End delay is defined as the average time taken for a data packet in order to reach the surface sink from the source node. Coverage is defined as each type of the heterogeneous node coverage and its relation to the total monitoring coverage of the network in different sizes of LSNs.

5.2 Comparison with other Addressing Schemes

Some addressing schemes are proposed for the pipeline monitoring process but they require geographical location details of each node that is still a challenging task for UW-LSN. H2-DAB [36] is a dynamic addressing scheme for UWSNs routing while ROLS [20, 43] is a static addressing scheme and its updated model [44] considers UW-LSNs routing issue which propose hexadecimal addresses for the heterogeneous types of nodes.

Comparing with H2-DAB:

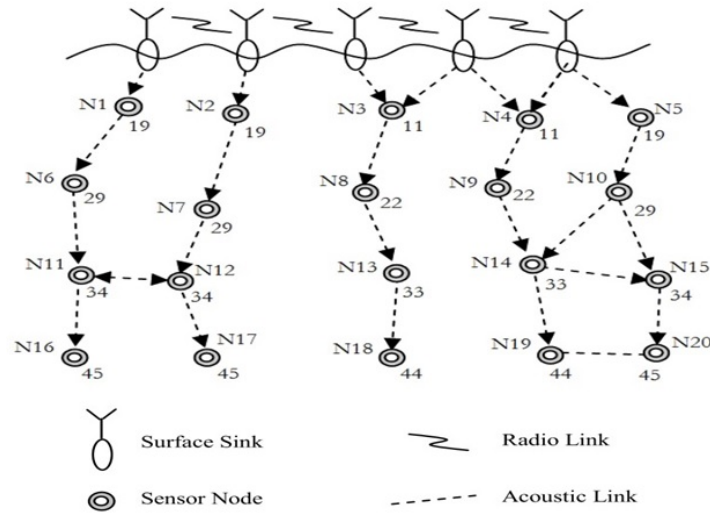


Fig. 8. H2-DAB Addressing Model [36]

H2-DAB [36] is the first dynamic addressing based UWSNs routing protocol which assigned a hop based address to each node in an efficient way without requiring any localization information its model shown in Fig. 8. H2-DAB addressing schemes develops its nodes addresses based on different number of hops in multi-sink architecture. While, H2-DARP-PM has some unique features such as deployment and addressing of heterogeneous types of nodes for LSN, topology distribution, providing of primary and secondary paths towards sinks and CN. The introduction of these features improved routing performance and minimize delay in packet forwarding of large scale UW-LSN. Following are some details about similarities and difference between H2-DAB and H2-DARP-PM.

Similarities:

- I. Both nodes address contains detail about number of hops.
- II. Multi-sinks architecture utilized by both protocols.
- III. CN used for data collection.

Differences/Uniqueness of H2-DARP-PM:

- I. H2-DARP-PM can assign addresses to heterogeneous types of nodes.
- II. H2-DARP-PM distributes network topology and especially designed for LSN.
- III. H2-DARP-PM considers CN as a secondary sink.

Comparing with ROLS:

ROLS [43] is a static addressing based LSN routing technique [20] that follows a hierarchical and multi-layer addressing model shown in Fig. 9. In ROLS address of each node consists of three fields which distinguish heterogeneous types of nodes. Although ROLS is efficient and considerably similar with H2-DARP-PM still it has some weaknesses such as missing of dynamic addressing, ignoring of hop based addressing, no CNs, topology distribution and multi-sink architecture which ultimately increase load and delay.

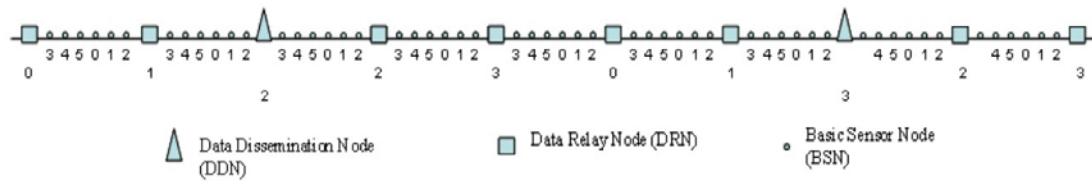


Fig. 9. ROLS Addressing Model [20]

Similarities:

- I. Heterogeneous types of nodes are utilized.
- II. Hierarchical network model topology utilized.
- III. Both node's Hop ID consist of the unique portion to maintain heterogeneity.

Differences/Uniqueness of H2-DARP-PM:

- I. H2-DARP-PM has dynamic addressing while ROLS has static addressing.
- II. H2-DARP-PM consider CNs while ROLS has a single sink and no any CN.
- III. H2-DARP-PM provide two paths towards the primary sink and secondary sink.

5.3 Simulation Results

Nodes Deployment and Network Coverage: Fig. 10(a) explains the frequency of heterogeneous nodes deployment in H2-DARP-PM for different sizes of pipelines. Keeping in view the lengths of the pipeline, we used 1000M pipeline and distributed its length into BSN, DRN, DDN and CN types of nodes. We have introduced CN and considered a higher frequency node which is deployed for monitoring of 1000M long pipeline. This frequency is calculated by the ratio of pipeline length and the range of each type of sensor. The range of each sensor is explained in Fig. 10(b) which is showing that the total number of nodes varies while we use different range sets for different types of nodes i.e Range Set 1(RS1)=BSN(100m), DRN(250m), DDN(400m) and CN(500m) ranges.

Comparing with Almazyad and ROLS:

In Fig. 10(c), we can observe the comparison of a number of nodes deployed in the homogeneous Almazyad deployment model [4] and heterogeneous H2-DARP-PM. The coverage of the network is always based on types of nodes like homogeneous or heterogeneous and their ranges. In this regard, Fig. 10(c) showing that the total number of heterogeneous nodes in a network can be varied according to their application requirement and range set while a total number of homogeneous nodes are always deployed at a fixed ratio like recommended by Almazyad nodes deployment model. In this model, homogenous types of nodes are deployed at different distances between 10M-500M but at a time one distance is maintained for all the nodes. The drawback of homogeneous nodes deployment is that it is always expensive because it requires more number of nodes and utilizes more resources, facing higher latency due to the usage of same types of nodes with fixed ranges. In the case of heterogeneous nodes, the delay issue could easily be overcome by setting the different ranges of sensors; besides, it also supports to distribute the topology into sub-lengths according to sensor's ranges. We have also compared the frequency of different types of nodes deployment in one monitoring segment of H2-DARP-PM with ROLS [43]. Fig. 10(d) shows that H2-DARP-PM has introduced CN not been used by any other pipeline monitoring routing technique including ROLS. ROLS model is only restricted to utilize the fixed number of BSNs

(6 nodes) while H2-DARP-PM is flexible enough to increase or decrease the number of nodes. Due to simulator limitations, we have deployed 4 BSNs per one DRN that help to monitor more area with minimum cost in the network.

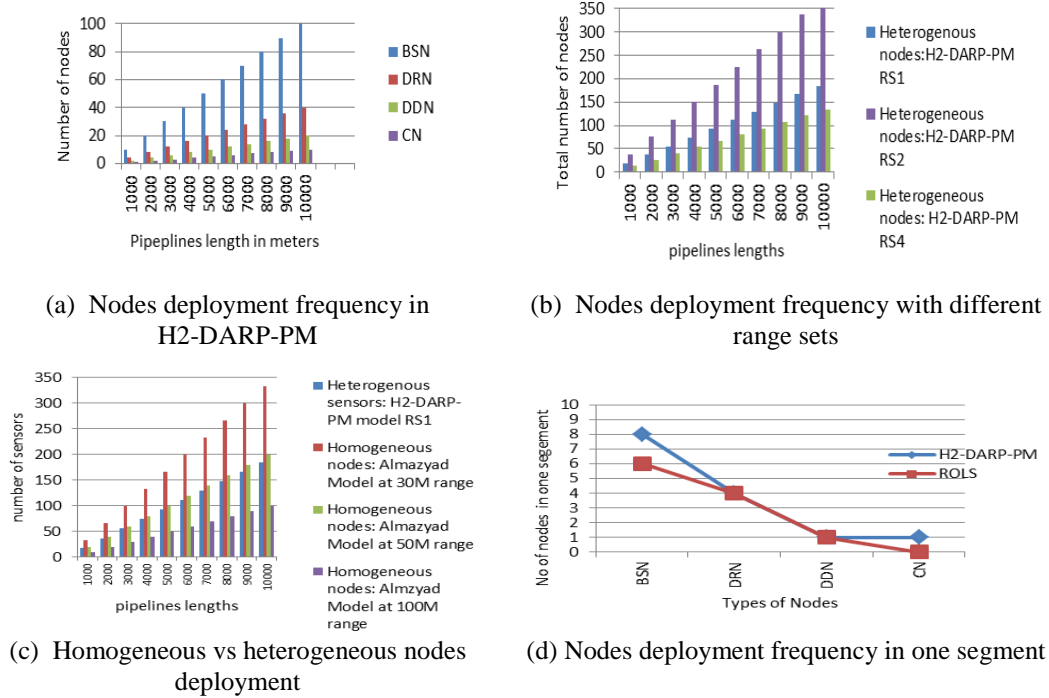
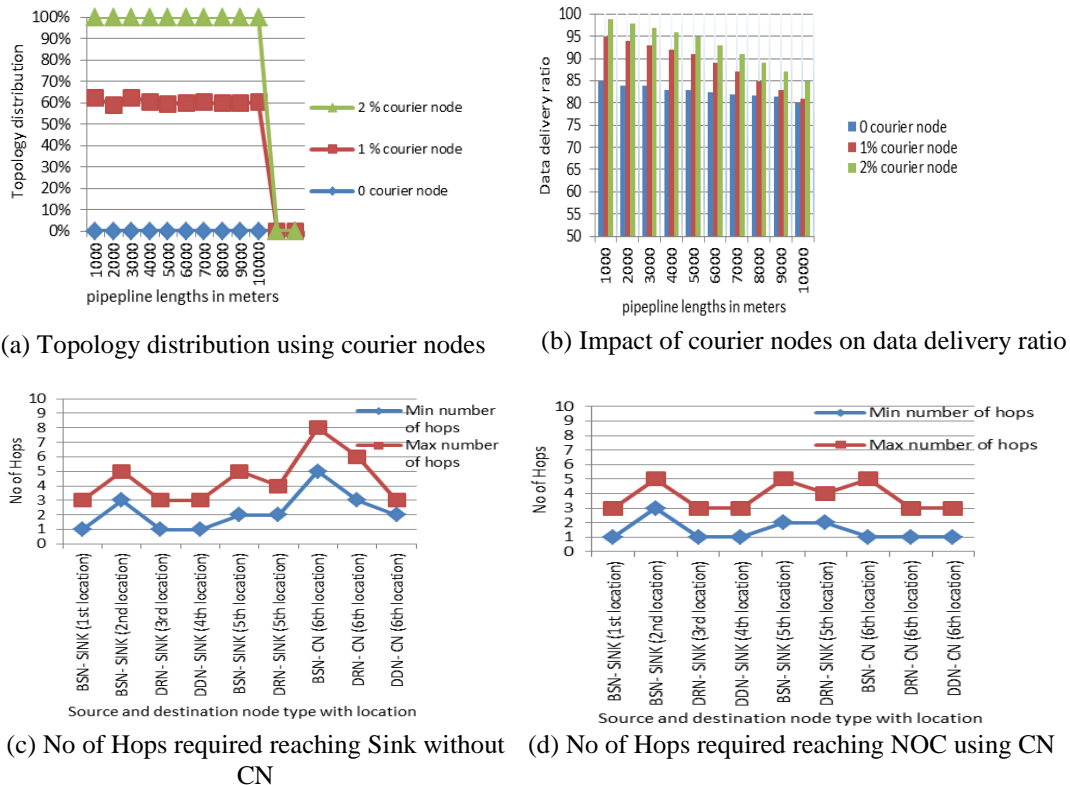


Fig. 10. The impact of nodes deployment on performance of H2-DARP-PM

The impact of Courier Nodes: H2-DARP-PM is the first routing technique that introduces CN to distribute the monitoring area. It supports to minimize the delay in packet forwarding process of nodes being far from the sinks. We have used one courier node in our proposed topology and approximately 50% of the network topology is distributed through its utilization. According to the **Fig. 11(a)** the impact of courier node remains higher in the topology distribution and minimization of delay.

In general, the higher delay has always had a serious effect on the data delivery and it grows more severe in the communication between underwater acoustic sensors. In the proposed topology, we have equally distributed the total communication range into two sinks deployed on both ends of the pipeline. In the case of no courier node in the network communication ranges, the medium level of data delivery ratio is achieved but the introduction of the courier node divides the communication area resulting increase in the data delivery ratio. Data delivery ratio may highly be affected by the introduction of 1 or 2 courier nodes in the case of small pipeline up to 1000M; while the same numbers of courier nodes do not have a major impact on 10000M long pipeline. There is still an opportunity to add more courier nodes in long range pipelines monitoring LSNs. H2-DARP-PM can complete its tasks without the presence of any courier node, but we suggest a small number of courier nodes for the topology distribution concerned and better resource utilization. **Fig. 11(b)** presents that how a different number of courier nodes help to increase the overall performance of the routing protocol. In **Fig. 11(a)** and **Fig. 11(b)** describe the topology distribution and data delivery ratio, we can see that H2-DARP-PM achieves 50% packet delivery ratio in small range pipelines without using

courier nodes; but in the case of long range pipelines monitoring, the presence of courier nodes lift these ratios. In such scenarios, the courier nodes easily accommodate the deficiency of the ordinary sensor nodes. From these results, it is evident that only 1-2% courier nodes improve data delivery ratios up to 85-95% with the presence of the ordinary nodes like BSN, DRN and DDN.



(a) Topology distribution using courier nodes (b) Impact of courier nodes on data delivery ratio

(c) No of Hops required reaching Sink without CN (d) No of Hops required reaching NOC using CN

X-Axis locations detail:

- 1st location=Before 1st DRN
- 2nd location =After 1st DRN and before 1st DDN
- 3rd location = Before 1st DDN
- 4th location =At 1st DDN
- 5th location = After first DDN
- 6th location = Near mid of the pipeline

Fig. 11. The effect of Courier nodes on H2-DARP-PM performance

Not only delivery ratio, the courier nodes also help to reduce the delay, computational overhead, maintaining of complex routing tables and energy consumption of ordinary sensor nodes. These courier nodes function like a secondary sink for collecting the data packets from middle area nodes of the pipeline whereas they move vertically in order to deliver these packets to the surface sinks directly. The introduction of courier nodes decreases the involvement of ordinary sensor nodes that also reduce the cost of per packet delivery ultimately increasing the life of the network. The minimum and a maximum number of hops required to reach sink nodes are shown in Fig. 11(c) and the number of hops varies from 1 to 8. Fig. 11(d) highlights the importance of using courier node in packet forwarding that proves helpful in minimizing the number of hops for transferring the data packets.

Offered Load Analysis: We have analyzed the performance of the H2-DARP-PM using different data delivery ratios and end-to-end delays. Usually, the sensor networks generate 1

packet per second, but here in our case, it generates different packets in different times such as 3 packets in every 2 seconds and likewise 2 packets per second. Fig. 12(a) presents the data delivery ratio with a different number of nodes. It shows that the delivery ratios remain almost the same with the more number of homogeneous nodes while it varies with the less number of nodes. At higher load with homogeneous nodes in the network, sometimes a node cannot find next hop due to the low data rate, collision, and conjunction or node failure resulting into overhead and packet loss. Fig. 12(b) shows the results of end-to-end delays when we gradually increase the number of packets in the network. It also highlights the ability of proposed network handling easily more number of packets; likewise, it can manage the delays up to three packets generated in the network.

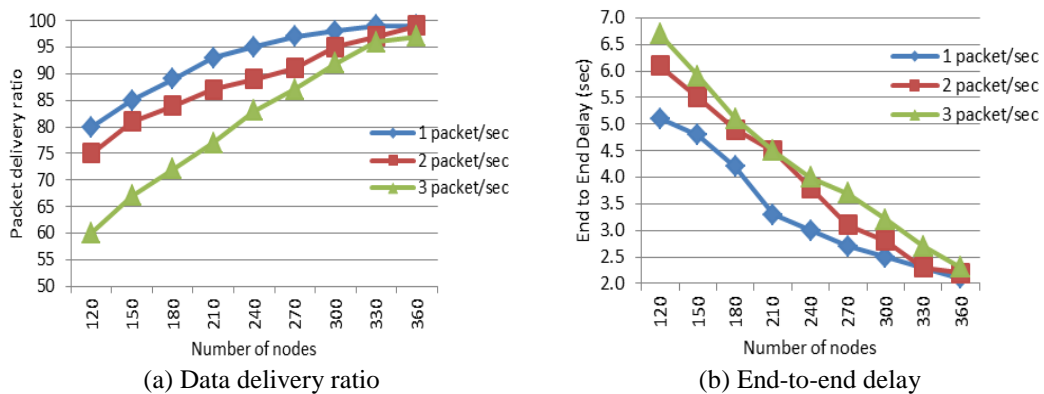


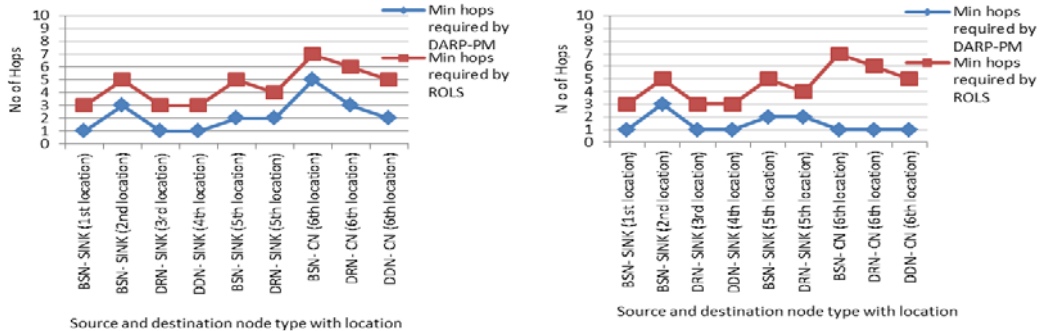
Fig. 12. The effect of different offered load on H2-DARP-PM performance

5.4 Comparison with H2-DAB and ROLSROLS

Few routing techniques are proposed for the pipeline monitoring process that requires geographical location details of each sensor node in the network. This type of problem is a challenging for UW-LSN. ROLS[20] considers localization issue and H2-DAB[36] considers underwater routing issue, both propose addressing schemes. ROLS assigns the hexadecimal addresses to the heterogeneous types of sensors while H2-DAB assigns binary notation addresses to homogeneous types of underwater sensors. When a node wants to forward the data packet, it calculates the neighbor nodes addresses and tries to target hierarchically higher level node. Similarly, the next forwarding node searches the next (upper) level node to forward the data packet; in the case of not finding the upper-level node, it searches for the same level node by using jump or redirect algorithm. This procedure is continued until the data packet reaches any of the sinks. However, ROLS involves some serious problems as compared to H2-DARP-PM and some of them are discussed. First ROLS requires that every node must have fixed number of neighbors and data packet strictly follow the hierarchical network model that not only increases the cost of the network but also results in higher latency issue. Secondly, ROLS can't handle the coverage issues for long range pipelines as it has not given any proper nodes deployment algorithm for the distributed topology length of the pipelines. Thirdly, it selects a single sink architecture that is not suitable for the monitoring of long-range underwater pipelines. Lastly, it does not elaborate nodes address initialization procedure and the packet forwarding process that how does it manage the neighbor nodes details and how does it select the best next hop forwarding node.

Further, we checked the performance of H2-DARP-PM in comparison to ROLS. First, we compared the minimum numbers of hops required by H2-DARP-PM and ROLS to transfer

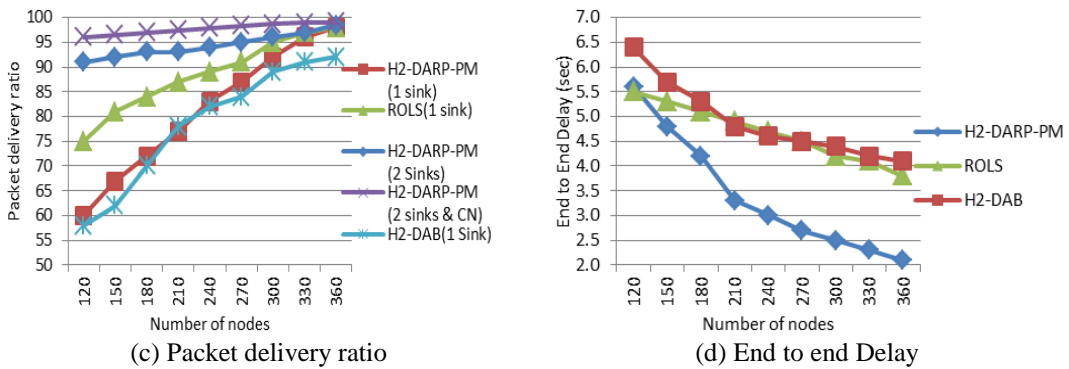
the data packet from different types of source nodes towards the Sink/CN. Fig. 13(a) presents the number of hops required by BSN, DRN or DDN nodes to transfer their data packets towards the sinks from different locations of the pipeline. It shows that ROLS need more hops to transfer its data packets. When we used courier nodes in H2-DARP-PM, they extremely minimized the number of hops required by middle area sensors for forwarding their data. The importance of CN is shown in Fig. 13(b).



(a) Number of hops required reaching primary sink

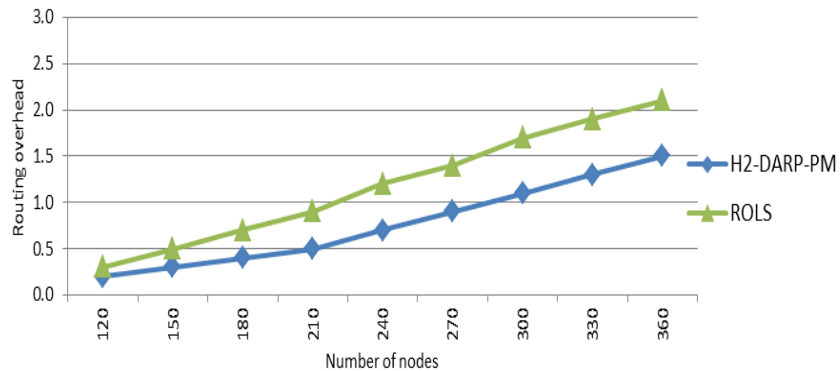
(b) Number of hops required to reach NOC using CN

Note: X-Axis details are same like given in Fig. 11.



(c) Packet delivery ratio

(d) End to end Delay



(e) Normalized routing overhead

Fig. 13. Performance comparison of H2-DARP-PM with ROLS

We have also compared the offered load of H2-DARP-PM with ROLS and H2-DAB at different delivery ratios with single and two sinks presented as shown in Fig. 13(c). Using 1sink H2-DAB presents lowest packet delivery ratio other then H2-DARP-PM and ROLS. In

this scenario, ROLS has better performance rather than H2-DARP-PM due to the deployment of a single sink at the top of pipeline mid area which divides 50% topology. On the other hand, while 2 sinks are deployed then H2-DARP-PM provides better packet delivery ratio even with a minimum number of nodes deployment. However, the deployment of fewer nodes decreases the delivery ratios in ROLS while it has less effect on H2-DARP-PM delivery ratios.

The main reason for the delay is that ROLS uses source-based greedy routing for data forwarding. In such routing, when no node is found with higher hierarchy level then it forwards data to the same level nodes that cause increased delay. Moreover, when we use single sink placed at the center area of the pipeline surface, we discover clear differences in the delivery ratios. Yet again due to the greedy routing of ROLS, BSNs try to send data packets towards the DRNs, and DRNs forward data packets towards DDNs. Further, the DDNs forwards them to the Sink/CN simultaneously. The ROLS and H2-DAB do not maintain the secondary route while the nodes of the upper-level broadcast the data packets. In this situation, there is a chance of no response from the neighbors due to nodes failures, low data rates and water currents. Due to these issues, data packets can be lost or discarded ultimately decreasing the delivery ratios.

Further, we have discussed the end-to-end delays in **Fig. 13(d)**; here H2-DARP-PM delivers data packets with less end-to-end delays while more numbers of sensor nodes are available in the deployed area. ROLS and H2-DAB take extra waiting time for all data packets which have been received by the ordinary nodes. ROLS also controls the duplication of packets when other neighboring nodes are also going to forward the same data packet. On the other hand, while the less number of nodes are deployed it causes a higher delay in H2-DAB and H2-DARP-PM. The delay occurs while distance increases between the nodes and frequency the neighbor nodes decreases. In this regards, while a node could not find the higher level next hop neighbor then it waits for a defined time and goes for 2nd or 3rd try. It results in minor level higher delays with less number of nodes deployed but ultimately it does not have a major impact on the delivery ratios.

Fig. 13(e) shows the normalized routing overhead with different network density. The H2-DARP-PM protocol significantly reduce the routing overhead incurred during the route discovery towards the next hop neighbour. Although the H2-DARP-PM protocol increases the packet size of inquiry reply packets but at the same time, it reduces the number of inquiry reply packets. Further, the inquiry request packet size is reduced significantly in H2-DARP-PM having only node ID. A node ID consists of all information regarding its neighbours; it helps to reduce the traffic load in estimating the shortest path towards the nearest sink or courier node. In order to obtain fairness in the results, the statistics of normalized routing overhead includes Hello traffic. The H2-DARP-PM protocol still yields improved normalized routing overhead performance. The routing overhead is negligible when less number of nodes are deployed decreasing the number of nodes in the neighbourhood of each node resulting into reduced traffic load. On the other hand, if increases the number of nodes in the same length of pipeline then nodes become closer to each other causing increase in the neighbour nodes. This phenomenon is explained **Fig. 13(e)**. ROLS has fixed neighbors and broadcasts packets at three different ranges for the selection of next hop. Similarly, it increases in total number of the nodes which greatly impacts on the increasing number of broadcast packets and traffic load. The proposed routing algorithm provides better results in most of the situations with different parameters. While ROLS and H2-DAB face problems in both situations i.e. when the number of nodes increases, the cost and energy consumption is high and when the numbers of nodes decreases, the packet delivery ratios are affected. On the other hand, H2-DARP-PM maintains very good delivery ratios with distributed topology and with a small number of nodes that

improves the monitoring coverage for the long range of underwater pipeline.

6. Important Aspects of the H2-DARP-PM

Beside the advantages such as low cost and requiring no dimensional location information for task completion, the following are some important aspects of H2-DARP-PM.

- **Monitoring coverage improved:** Heterogeneous types of nodes are used to cover the pipeline length. The total length of pipeline is divided according to the range of each type of node deployed at the specific position. This deployment method is dynamic in nature; if pipeline length increases then it can easily assign relative positions to new the nodes.
- **Easy to find the damage/leakage position:** In H2-DARP-PM, every node has a unique address which maintains its neighbor nodes details. The nodes are deployed sequentially so that it could be easily determined that which node has detected damage or leakage of the pipeline. From node's address, it is easy to find the exactly damaged location of the pipeline. H2-DARP-PM is highly robust in a way that nodes may become part of a network or leave a network smoothly without affecting the major causes.
- **Easy to focus critical areas of the pipeline:** H2-DARP-PM allows adding more sensors on critical areas or most damaged areas of the pipeline. Most of the existing protocols consider pipeline as a single unit while proposed protocol distributes the pipeline length into sub-lengths according to the ranges of higher frequency sensors. This topology model also helps to focus more on critical portions.
- **Collision free routing:** As during the routing process, collision stays common in dynamic addressing based protocols. However, H2-DARP-PM distributes the network topology, assign dynamic addresses and control communication process intelligently in order to avoid the occurrence of collisions.
- **The problem of table size:** The size of the routing table is one of the critical factors in dynamic addresses schemes. However, in our proposed H2-DARP-PM, the size of the routing table is limited as it maintains neighbor nodes details and routing decision based on HopID and level of the neighbor nodes. Further, the size of the network does not have any major impact on the routing table.
- **The problem of address space exhaustion:** Most of the dynamic addressing schemes are constraints in terms of limited address space; however our proposed scheme do not face this problem due to having fixed address space till 8 digits per HopID.
- **Higher latency problem:** Underwater applications mostly use acoustic communications that happen to be five times slower than RF; therefore, it is important to minimize the delay. Considering this issue, H2-DARP-PM tries to minimize the delay by using heterogeneous types of nodes with communication ranges.
- **Minimization of computational overhead:** Normally routing process creates a high level of computational overheads due to the exchange of many control packets for routing decisions that are not acceptable in acoustic communication. H2-DARP-PM use an only 8-bit address where each node can calculate and select next hop by minor computation and forward data packets without sharing many control packets.
- **Monitoring areas at different depths:** In most of the underwater pipeline monitoring applications, the underwater monitored area consists of coastal areas and shallow water

with low-level depth. On the other hand, H2-DARP-PM is not designed for a specific area; it can work at any depth level as it has mainly concern with the pipeline sensors.

7. Conclusions and Future Work

In this paper, we have proposed a dynamic addressing based routing protocol using linear sensor networks based on heterogeneous types of sensors and hop-by-hop addressing for underwater pipeline monitoring. The novelty of this protocol involves its efficient monitoring coverage with minimum delay including no requirement of maintaining complex routing tables of entire network. Since the underwater acoustic communication supports extremely low data rates, so we have minimized the delay. In this research, we used the idea of per-contact routing instead of source routing or per-hop routing according to the underwater conditions. An important aspect of H2-DARP-PM is that the packet delivery ratios are not based on the density or sparseness of sensor nodes rather on the distributed topology and dynamic addressing. The findings reveal that although minimization of delay stays a challenge for underwater routing but it is handled efficiently in H2-DARP-PM. The problem of node failure being a major threat is handled in the proposed protocol by using the jumping strategy. Further revelations indicate that H2-DARP-PM deployment remains the robust and scalable for the new nodes whenever added at relative deployment location; these new nodes also get addresses accordingly. Finally, H2-DARP-PM is flexible enough to be used for the long-term pipeline monitoring applications; it even has the capability to cover the long range pipelines using less number of nodes with no any limitation of the network size. In future, we look up to integrate H2-DARP-PM with several underwater MAC protocols in order to further investigate the reliability of the network, failure nodes detection, route maintenance and energy efficiency relative performance.

Acknowledgment

We thank MOHE Malaysia and COMSATS Pakistan for supporting this research program. Authors also thank anonymous reviewers and Mr Fasee Ullah for assistance and comments that greatly improved the manuscript quality.

References

- [1] S. Varshney, C. Kumar, and A. Swaroop, "Linear sensor networks: Applications, issues and major research trends," in *Proc. of Computing, Communication & Automation (ICCCA), 2015 International Conference on*, pp. 446-451, 2015. [Article \(CrossRef Link\)](#)
- [2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, pp. 257-279, 2005. [Article \(CrossRef Link\)](#)
- [3] M. Murad, A. A. Sheikh, M. A. Manzoor, E. Felemban, and S. Qaisar, "A Survey on Current Underwater Acoustic Sensor Network Applications," *International Journal of Computer Theory and Engineering*, vol. 7, p. 01, 2015. [Article \(CrossRef Link\)](#)
- [4] A. S. Almazyad, Y. M. Seddiq, A. M. Alotaibi, A. Y. Al-Nasheri, M. S. BenSaleh, A. M. Obeid, *et al.*, "A Proposed Scalable Design and Simulation of Wireless Sensor Network-Based Long-Distance Water Pipeline Leakage Monitoring System," *Sensors*, vol. 14, pp. 3557-3577, 2014. [Article \(CrossRef Link\)](#)
- [5] S. Ali, S. B. Qaisar, H. Saeed, M. F. Khan, M. Naeem, and A. Anpalagan, "Network Challenges for Cyber Physical Systems with Tiny Wireless Devices: A Case Study on Reliable Pipeline Condition Monitoring," *Sensors*, vol. 15, pp. 7172-7205, 2015. [Article \(CrossRef Link\)](#)

- [6] N. Mohamed, I. Jawhar, J. Al-Jaroodi, and L. Zhang, "Sensor network architectures for monitoring underwater pipelines," *Sensors*, vol. 11, pp. 10738-10764, 2011. [Article \(CrossRef Link\)](#)
- [7] J.-H. Kim, G. Sharma, N. Boudriga, and S. S. Iyengar, "SPAMMS: a sensor-based pipeline autonomous monitoring and maintenance system," in *Proc. of Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, pp. 1-10, 2010. [Article \(CrossRef Link\)](#)
- [8] A. A. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii, "A New Approach to the Sewer Pipe Inspection: Fully Autonomous Mobile Robot" KANTARO", in *Proc. of IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pp. 4088-4093, 2006. [Article \(CrossRef Link\)](#)
- [9] T. T.-T. Lai, W.-J. Chen, K.-H. Li, P. Huang, and H.-H. Chu, "TriopusNet: automating wireless sensor network deployment and replacement in pipeline monitoring," in *Proc. of the 11th international conference on Information Processing in Sensor Networks*, pp. 61-72, 2012. [Article \(CrossRef Link\)](#)
- [10] A. Nasr and A. Courbot, "A new approach to Pipeline Inspection using Autonomous Underwater Vehicles (AUV's)," 2013. [Article \(CrossRef Link\)](#)
- [11] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "An efficient framework for autonomous underwater vehicle extended sensor networks for pipeline monitoring," in *Proc. of Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on*, pp. 124-129, 2013. [Article \(CrossRef Link\)](#)
- [12] C. W. Chen and Y. Wang, "Chain-Type Wireless Sensor Network for Monitoring Long Range Infrastructures: Architecture and Protocols*," *International Journal of Distributed Sensor Networks*, vol. 4, pp. 287-314, 2008. [Article \(CrossRef Link\)](#)
- [13] R. Shokri, N. Yazdani, and A. Khonsari, "Chainbased anonymous routing for wireless ad hoc networks," in *Proc. of the 4th IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 297-302, 2007. [Article \(CrossRef Link\)](#)
- [14] I. Stoianov, L. Nachman, S. Madden, T. Tokmouline, and M. Csail, "PIPENET: A wireless sensor network for pipeline monitoring," in *Proc. of Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pp. 264-273, 2007. [Article \(CrossRef Link\)](#)
- [15] J. Kim, J. S. Lim, J. Friedman, U. Lee, L. Vieira, D. Rosso, *et al.*, "SewerSnort: A drifting sensor for in-situ sewer gas monitoring," in *Proc. of Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*, pp. 1-9, 2009. [Article \(CrossRef Link\)](#)
- [16] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proc. of the 1st Annual conference on Research in information technology*, pp. 51-56, 2012. [Article \(CrossRef Link\)](#)
- [17] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, and C. Shahabi, "SWATS: Wireless sensor networks for steamflood and waterflood pipeline monitoring," *Network, IEEE*, vol. 25, pp. 50-56, 2011. [Article \(CrossRef Link\)](#)
- [18] I. Jawhar, N. Mohamed, and L. Zhang, "A distributed topology discovery algorithm for linear sensor networks," in *Proc. of Communications in China (ICCC), 2012 1st IEEE International Conference on*, pp. 775-780, 2012. [Article \(CrossRef Link\)](#)
- [19] X. Sun, J. He, Y. Chen, S. Ma, and Z. Zhang, "A new routing algorithm for linear wireless sensor networks," in *Proc. of Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on*, pp. 497-501, 2011. [Article \(CrossRef Link\)](#)
- [20] I. Jawhar, N. Mohamed, K. Shuaib, and N. Kesserwan, "An Efficient Framework and Networking Protocol for Linear Wireless Sensor Networks," *Adhoc & Sensor Wireless Networks*, vol. 7, 2009. [Article \(CrossRef Link\)](#)
- [21] H. Yu and M. Guo, "An efficient oil and gas pipeline monitoring systems based on wireless sensor networks," in *Proc. of Information Security and Intelligence Control (ISIC), 2012 International Conference on*, pp. 178-181, 2012. [Article \(CrossRef Link\)](#)

- [22] D. Wu, K. Youcef-Toumi, S. Mekid, and R. B. Mansour, "Relay node placement in wireless sensor networks for pipeline inspection," in *Proc. of 2013 American Control Conference*, pp. 5905-5910, 2013. [Article \(CrossRef Link\)](#)
- [23] L. Qiu, I. Kevin, K. Wang, and Z. Salcic, "Dynamic duty cycle-based Wireless Sensor Network for underground pipeline monitoring," in *Proc. of 2015 9th International Conference on Sensing Technology (ICST)*, pp. 116-121, 2015. [Article \(CrossRef Link\)](#)
- [24] K. Anupama, N. Kamdar, S. K. Kamalampet, D. Vyas, S. Sahu, and S. Shah, "A wireless sensor network based pipeline monitoring system," in *Proc. of Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*, pp. 412-419, 2014. [Article \(CrossRef Link\)](#)
- [25] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks," *Journal of Intelligent & Robotic Systems*, vol. 74, pp. 437-453, 2014. [Article \(CrossRef Link\)](#)
- [26] V. Vinoth, M. Muthaiah, and M. Chitra, "An efficient pipeline inspection on heterogeneous relay nodes in wireless sensor networks," in *Proc. of Communications and Signal Processing (ICCSP), 2015 International Conference on*, pp. 0730-0734, 2015. [Article \(CrossRef Link\)](#)
- [27] A.-K. Tariq, A.-T. Ziyad, and A.-O. Abdullah, "Wireless sensor networks for leakage detection in underground pipelines: a survey paper," *Procedia Computer Science*, vol. 21, pp. 491-498, 2013. [Article \(CrossRef Link\)](#)
- [28] G. Owojaiye and Y. Sun, "Focal design issues affecting the deployment of wireless sensor networks for pipeline monitoring," *Ad Hoc Networks*, vol. 11, pp. 1237-1253, 2013. [Article \(CrossRef Link\)](#)
- [29] J. Butkus, L. Jakevičius, and O. Tumšys, "An acoustic method of determination of leakage coordinates in gas pipelines," *Archives of Acoustics*, vol. 23, pp. 533-540, 2014. [Article \(CrossRef Link\)](#)
- [30] M. Lingya, L. Yuxing, S. Liqun, Z. Fangsheng, and F. Juntao, "Acoustic propagation characteristics, position monitoring and locating of gas transmission pipeline leakage [J]," *Natural Gas Industry*, vol. 11, p. 023, 2010. [Article \(CrossRef Link\)](#)
- [31] A. Mahfurdz, S. Sunardi, H. Ahmad, S. Abdullah, and N. Nazuki, "Distinguish Sea Turtle and Fish Using Sound Technique in Designing Acoustic Deterrent Device," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 13, pp. 1305-1311, 2015. [Article \(CrossRef Link\)](#)
- [32] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile Ad Hoc networking: the cutting edge directions* vol. 35: John Wiley & Sons, 2013. [Article \(CrossRef Link\)](#)
- [33] D. Inaudi and B. Glisic, "Long-range pipeline monitoring by distributed fiber optic sensing," *Journal of pressure vessel technology*, vol. 132, p. 011701, 2010. [Article \(CrossRef Link\)](#)
- [34] I. Jawhar, N. Mohamed, and D. P. Agrawal, "Linear wireless sensor networks: Classification and applications," *Journal of Network and Computer Applications*, vol. 34, pp. 1671-1682, 2011. [Article \(CrossRef Link\)](#)
- [35] M. Ayaz and A. Abdullah, "Hop-by-hop dynamic addressing based (H2-DAB) routing protocol for underwater wireless sensor networks," in *Proc. of Information and Multimedia Technology, 2009. ICIMT'09. International Conference on*, pp. 436-441, 2009. [Article \(CrossRef Link\)](#)
- [36] M. A. A. Arshad, "EFFICIENT DYNAMIC ADDRESSING BASED ROUTING FOR UNDERWATER WIRELESS SENSOR NETWORKS," UNIVERSITI TEKNOLOGI PETRONAS, 2011. [Article \(CrossRef Link\)](#)
- [37] M. Z. Abbas, K. A. Bakar, M. A. Arshad, M. Tayyab, and M. H. Mohamed, "Scalable Nodes Deployment Algorithm for the Monitoring of Underwater Pipeline," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, 2016. [Article \(CrossRef Link\)](#)

- [38] M. S. BenSaleh, S. M. Qasim, A. M. Obeid, and A. Garcia-Ortiz, "A review on wireless sensor network for water pipeline monitoring applications," in *Proc. of Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pp. 128-131, 2013. [Article \(CrossRef Link\)](#)
- [39] N. Mohamed, J. Al-Jaroodi, I. Jawhar, and A. Eid, "Reliability Analysis of Linear Wireless Sensor Networks," in *Proc. of Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, pp. 11-16, 2013. [Article \(CrossRef Link\)](#)
- [40] M. Ayaz and A. Abdullah, "Underwater wireless sensor networks: routing issues and future challenges," in *Proc. of the 7th International Conference on Advances in Mobile Computing and Multimedia*, pp. 370-375, 2009. [Article \(CrossRef Link\)](#)
- [41] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, pp. 1908-1927, 2011. [Article \(CrossRef Link\)](#)
- [42] S. H. Syed Ariffin, N. S. N. Ismail, and L. A. Hussein, "Analyzing the performance of acoustic channel in Underwater Wireless Sensor Network (UWSN)," 2010. [Article \(CrossRef Link\)](#)
- [43] I. Jawhar, N. Mohamed, M. M. Mohamed, and J. Aziz, "A routing protocol and addressing scheme for oil, gas, and water pipeline monitoring using wireless sensor networks," in *Proc. of Wireless and Optical Communications Networks, 2008. WOCN'08. 5th IFIP International Conference on*, pp. 1-5, 2008. [Article \(CrossRef Link\)](#)
- [44] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "Data communication in linear wireless sensor networks using unmanned aerial vehicles," in *Proc. of Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pp. 492-499, 2013. [Article \(CrossRef Link\)](#)



Muhammad Zahid Abbas received his B.Sc. degree in Mathematics and Physics from Bahauddin Zakariya University Multan, Pakistan in 2000. In 2004, he received his Master degree in Computer Science from the Punjab University college of information technology(PUCIT) at University of the Punjab, Lahore, Pakistan. In 2008, he received his MS degree in Computer Science from Uppsala University, Uppsala, Sweden. From 2011 he is serving as a lecturer in the Computer Science Department at COMSATS Institute of Information and Technology, Pakistan. Currently, he is doing his PhD at Universiti Teknologi Malaysia (UTM), Malaysia under the kind supervision of Professor Dr. Kamalrulnizam Abu Bakar. His research interest includes the areas of Routing and Monitoring in Wireless Sensor Networks, UWSN, WMN and VANET.



Kamalrulnizam Abu Baker is a professor in Computer Science at Universiti Teknologi Malaysia (UTM), Malaysia, and a member of the Pervasive Computing research group. He received his B.Sc. degree in computer science from the Universiti Teknologi Malaysia (UTM), Malaysia, in 1996, M.Sc. degree in computer communications and Networks, from Leeds Metropolitan University, United Kingdom, in 1998, and his Ph.D. degree in computer science from Aston University, United Kingdom, in 2004. His research interest includes mobile and wireless computing, ad hoc and sensor networks, information security and grid computing. He is a member of ACM and International Association of Engineering. He involves in many research projects and is a referee for several scientific journals and conferences.



Muhammad Ayaz received his PhD degree in Information Technology from Universiti Teknologi PETRONAS, Malaysia, and MS degree in Computer Sciences from SZABIST Islamabad Pakistan in 2011 and 2007 respectively. During MS, his specialization area belongs to Networks and Communication. He worked as Assistant Professor at Department of Computer Sciences of Federal Urdu University of Arts Science and Technology, Islamabad, Pakistan during 2007-08. His research interests include Mobile and Sensor Networks, Routing Protocols, Network Security and Underwater Acoustic Communications. He has published numerous papers in highly reputed international conferences and journals. One of his research article “A Survey on Routing Techniques in Underwater Wireless Sensor Networks” listed among the Top-Cited articles under Science-direct database. Additionally, he is active reviewer of many reputed journals belongs to IEEE Transactions, Elsevier and Springer. Currently he is serving department of Computer Engineering as Assistant Professor, also member of Sensor Networks and Cellular Systems (SNCS) research center, the University of Tabuk, KSA.



Mohammad Hafiz Mohamed received his B.Sc degree in Computer Science from Universiti Teknologi Malaysia (UTM), Malaysia, in March 2009. He is currently pursuing a doctoral degree at the Faculty of Computing in Universiti Teknologi Malaysia (UTM). His current research interests include underwater acoustic communication and networking, and sensor node localization for underwater wireless sensor networks.



Moeenuddin Tariq received his Master degree in computer science from Comsats Institute of Information Technology (CIIT), Wah Cantt, Pakistan in 2004 and M.S. degree in communication networks from Mohammad Ali Jinnah University Islamabad, Pakistan in 2008. In Nov 2016, he successfully completed his PhD degree in Computer Sciences at University Teknologi Malaysia. He has previously worked as Access Engineer in Nayatel. Pvt. Ltd. an optical fiber based ISP in Islamabad Pakistan from April 2009 to October 2010. He has also worked as a Lecturer from 2011 to 2013 in Federal Urdu University, Islamabad, Pakistan. Mr. Tariq currently has few quality research publications in the field of underwater networks. His research interest include underwater sensor network routing protocols and mobile ad-hoc networks.