Computer Communications 210 (2023) 147–162

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A novel routing protocol for underwater wireless sensor networks based on shifted energy efficiency and priority

Muhammad Ismail [a], Hamza Qadir [a], Farrukh Aslam Khan [b], Sadeeq Jan [a], Zahid Wadud [a], Ali Kashif Bashir [c,d,*]

[a] Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Peshawar 25000, Pakistan
[b] Center of Excellence in Information Assurance, King Saud University, Riyadh 11653, Saudi Arabia
[c] Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M15 6BH, UK
[d] Woxsen School of Business, Woxsen University, Hyderabad, India

ARTICLE INFO

Keywords:
Underwater Wireless Sensor Networks (UWSNs)
Energy efficiency
Routing protocol
Internet of Things (IoT)

ABSTRACT

Underwater Wireless Sensor Networks (UWSNs) are among the most promising research areas these days due to their unique characteristics and diverse underwater applications. Though a number of routing protocols have been designed and implemented for UWSNs over the past few years, the researchers face several challenges, e.g., low speed of propagation, small bandwidth, limited battery power, etc., while designing routing protocols for communication in UWSNs. Acoustic sensor nodes are equipped with batteries with limited power and it is quite costly to replace or recharge them. The network will not survive for the desired period of time if the power of node batteries is not efficiently used. To effectively resolve this issue, this paper proposes *a Shifted Energy Efficiency and Priority (SHEEP)* routing protocol for UWSNs. The proposed protocol aims to enhance the efficiency of the state-of-the-art *Energy Balanced Efficient and Reliable Routing* ($EBER^2$) protocol for UWSNs. *SHEEP* is built upon the depth and energy of the current forwarding node, the depth of the expected next forwarding node, and the average energy difference among the expected forwarders. Simulation results demonstrate that *SHEEP* improves the energy efficiency and packet delivery ratio in comparison to $EBER^2$ by 7.4% and 13% respectively.

1. Introduction

Underwater Wireless Sensor Networks (UWSNs) are growing tremendously with a potential of offering a large number of schemes for optimizing communication among the sensor nodes [1–3]. UWSNs can be used for emerging marine technologies, such as bathymetry (depth measurement of the ocean), exploration of geological resources (e.g., gas, oil, etc.), tracking or detection of fishing banks, maritime archaeology, and so on [4–6]. Researchers during the 90's became aware of some new features that were applicable to the UWSN communication and since then, UWSNs are among the main focus of study for numerous researchers working in this area [7–13].

The terrestrial wireless networking technologies, such as the Internet of Things (IoT) and 5G/6G networks, have received a considerable amount of attention during the past 15 years, not just in the area of standardization, but also in the market deployment of a certain number of devices, services, and applications [14–17]. Among these advanced products, Wireless Sensor Networks (WSNs) exhibit a spectacular boom as being one of those technologies that have a significant impact on industrial and scientific development [18–22]. Lately, UWSNs were proposed to be deployed underwater, where many of their applications like monitoring the water pollution, water quality testing, aquaculture, oil explorations, etc., can benefit from this highly effective technology. Despite similar functionalities, UWSNs also have some architectural differences with terrestrial wireless networks due to the transmission channel characteristics (water) and the acoustic ultrasound signals (signals employed for data transmission). Unfortunately, the design considerations of UWSNs are quite challenging due to the aquatic conditions of the communication system. As a result, the methods applied for terrestrial WSNs are not fully applicable to UWSNs. Hence, a general reassessment of the whole network is a must so as to supply optimal network services for a particular application's demand. Underwater nodes can communicate using acoustic links [23,24], radio or electromagnetic signals [25,26], or optical communication [27,28]. Among them, the acoustic communication is more suitable for UWSNs; however, the researchers face several challenges in acoustic communication. Some of the major challenges include:

* Corresponding author.
E-mail address: dr.alikashif.b@ieee.org (A.K. Bashir).

https://doi.org/10.1016/j.comcom.2023.07.014
Received 1 July 2021; Received in revised form 2 July 2022; Accepted 10 July 2023
Available online 18 July 2023

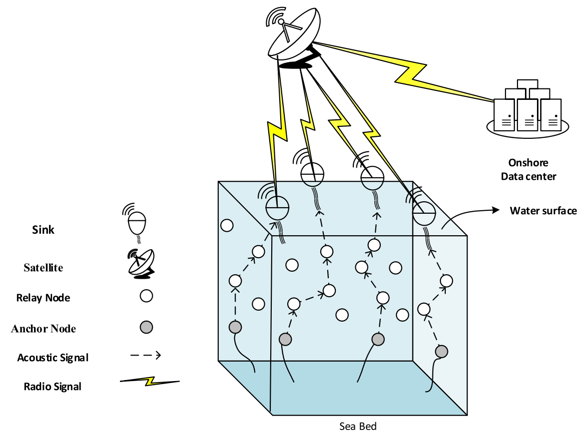

Fig. 1. Network architecture of UWSNs.

- Limited battery power (the batteries cannot be recharged; exploiting solar power is not possible)
- Availability of limited bandwidth
- The medium suffers from long propagation delays, fading, and multipath issues
- Very high rate of bit errors [29,30]
- Because of corrosion and fouling, the underwater sensors face frequent failures

UWSNs consist of a group of devices called nodes. They are the main elements of UWSNs and can sense and communicate with the other nodes. These nodes can be categorized into three main types: anchor nodes, relay nodes, and sink nodes, as depicted in Fig. 1. Normally, anchor nodes are nodes that know their location information [31]. However, here the anchor nodes refer to those attached with the chord that is further connected to the anchor. These nodes have several properties including sensing, communication, and storage. These nodes forward the (sensed) data to the higher layer consisting of relay nodes.

Relay nodes are deployed at different positions for sensing purposes and to forward or route all the network data. Since these nodes can move with the water current, their position is usually not fixed. Relay nodes work as a bridge between the lower layer and the surface sinks of the network. Sink nodes are the onshore stations that gather data from the entire remaining network. Anchor and relay nodes are embedded with an acoustic modem only, while both acoustic as well as radio modems are present in sink nodes. Sink nodes have to communicate with both underwater nodes as well as offshore stations. They work like a bridge between underwater nodes and offshore stations; they forward the received data from the underwater area to the offshore stations using radio links.

Many routing protocols and schemes are discussed in the literature that aim to design efficient communication techniques for UWSNs while considering underwater communication issues, e.g., increased bit error rate, high transmission and reception energy, low communication bandwidth, etc., at the same time. There exist various categories of routing protocols for UWSNs; however, we classify them into two main categories, i.e., location-based routing protocols and depth-based routing protocols. In the first category, every single node knows its own location, the target (destination), and sometimes the neighbors' location [32–36]. These protocols operate on the principal of directional forwarding, where packets are forwarded towards their destinations based on their direction for minimizing the propagation delay, reducing the energy consumption, and utilizing low bandwidth via making restrictions on broadcasting duplicate packets. On the other hand, no location information is needed for depth-based routing and only the information about the node's depth is required [37–42]. Estimation of depth does not need exchanging control packets but it can be calculated locally by using pressure sensors embedded in the node. Depth-based routing minimizes propagation delay by selecting a node among neighbors having minimum depth compared to the originator. All the remaining nodes between the sender and the forwarder simply discard the packet [37]. The packet suppression at the intermediate nodes is made possible by estimating the holding time for each node.

Once a data packet is received at a node, the depth information (embedded in the packet) is checked. If it is smaller than the sender's depth, then the node calculates the holding time based on depth, otherwise the node discards the packet. This strategy limits a large amount of duplicated transmissions, leading to lower energy consumption of the nodes. The node with the smallest depth must have the shortest holding time so that the closest node to the destination is selected for forwarding the packet. After the holding time estimation, the node initiates a timer. Upon the expiry of the holding time, the node checks whether it has received a copy of the packet. In case no duplicate copy has been received, the packet is further forwarded. Otherwise, the packet transmission is suppressed and the buffer is emptied. Some routing schemes also consider other metrics, e.g., link quality, node's energy, and depth of the next hop forwarder, etc. [38–42].

In depth-based forwarding schemes, the node with the smallest depth may deplete its energy earlier because all the packets are forwarded through the same set of nodes. Due to this problem, void regions are created in the network at the edge nodes failing to communicate further with the destination. To avoid packet transmission through the edge nodes of the void regions, some depth-based routing schemes take into account the depth difference of the sender/receiver and the neighbors of the receiver that are at a distance of one-hop. These types of schemes ensure the availability of potential forwarders for the receiver nodes. However, they fail to avoid the formation of void regions and cannot achieve energy balance in the network. To improve the performance of such types of schemes, strong balancing techniques are required to evenly distribute data traffic so that a single node is not used again and again for packet forwarding. In this way, the potential formation of the void region can be delayed.

To address the above issues, in this paper, we propose a routing scheme called *Shifted Energy Efficiency and Priority (SHEEP)*. *SHEEP* considers the depth difference between the sender, receiver, and one-hop neighbors of the receiver nodes.

1.1. Contributions

The main contributions of this work include:

- **Avoidance of Void Region Formation**: *SHEEP* does not select a single node repeatedly to avoid the void region formation.
- **Balanced Energy Consumption**: Energy consumption of the network is balanced as the best forwarder is the node that has the highest remaining residual energy.
- **Reduced Packet Duplication**: *SHEEP* selects a node from a region that has the highest average energy difference among the neighbors to avoid packet duplication.
- **Increased Average Operational Time:** The average operational time of the network is increased because of postponing the death of the network due to effective utilization of energy.
- **Reduced End-to-End-Delay (E2ED)**: The E2ED is reduced because of maintaining shorter path to the sink for longer duration due to higher operational time.

The remainder of the paper is organized as follows: Section 2 describes the related work and summarizes it in a comparison table (Table 1). Section 3 contains the detailed description about the problem formulation, proposed protocol, and holding time calculation of *SHEEP*. Section 4 is the experimental part that discusses the simulation results and compares the performance of *SHEEP* with the existing routing protocols. Finally, the conclusions and future directions are presented in Section 5.

Table 1
The existing routing protocols for underwater sensor networks.

Protocol	Routing mechanism	Achievements	Drawbacks
DBR [37]	Depth based	Relatively decreased unnecessary flooding	Minimum depth nodes are always penalized and die out early
EEDBR [38]	Depth and Energy based	Enhances PDR by considering residual energy of a forwarder in calculating the holding time	Fails to control duplicate packets especially in dense network scenario
WDFAD-DBR [42]	Depth based	High PDR is achieved by considering depth of next hop forwarder in a sparse network; less energy consumption and minimum end-to-end delay	Fails to balance network traffic; nodes of minimum depth are penalized, which lead to premature death of the network
LLSR [39]	Depth and link quality based	Considers path quality while forwarding the packet	No strategy for energy balancing; overhead produced due to large number of control packets
DOW-PR [43]	Depth and link quality based	Reduces energy consumption by selecting forwarder of minimum possible hops towards the sink	Lacks technique for energy balancing and sometimes minimum depth nodes are penalized
$EBER^2$ [44]	Depth and energy based	Does not penalize minimum depth nodes by considering residual energy	Not considering average energy difference among the neighbors on the second hop
RPSOR [45]	Depth, energy, and link quality based	Increased reliability and decreased duplicate packets by intelligent design of holding time	Fails to select the forwarder from a region where average energy difference among the neighbors is high

2. Related work

Over the past few years, several routing protocols have been proposed to make the underwater communication more efficient and reliable. To reduce duplicate packets, some routing schemes use holding time to decide the node for forwarding the packet; we call it receiver-based routing. The node with the shortest holding time gets the highest priority to forward the packet, as discussed in [37,38,40], and [42]. Some routing schemes allow the sender to decide the forwarder in advance; the packet header contains the information of the forwarder, i.e., to which node the packet should be forwarded, as done in [39,41]. We call it sender-based routing. In the former case, the receiver decides the forwarding node, while in the later case, the sender decides. Since our proposed routing protocol is receiver-based, we only focus on receiver-based routing protocols in this section.

In [37], the authors proposed Depth-Based Routing (DBR), a receiver-based scheme that depends on the depth information of the corresponding nodes in UWSNs. This scheme uses a greedy approach for delivering data packets towards the sink nodes at the water surface. Unlike geographical routing, which requires full-dimensional location information of the sensor nodes, this scheme requires the depth information only, which is quite easy to get. Moreover, this scheme can make use of multiple sinks without any extra cost. The simulation results showed that the DBR protocol improved the Packet Delivery Ratio (PDR) up to 95 percent in case of dense networks with sound consumption of energy. However, DBR fails to balance energy among the neighbors; in other words, it fails to balance data traffic among the neighbors. Due to this low depth, nodes always deplete early and the network cannot survive for the desired period of time. In addition, it generates many duplicate packets, especially in a dense network, which give rise to high energy consumption and packet collision.

Authors in [38] proposed a receiver-based routing scheme called Energy Efficient Depth Based Routing (EEDBR); a scheme which enhances the forwarding strategy of DBR. EEDBR considers residual energy in addition to the depth of the sensor nodes in computing the holding time. Every node maintains a neighbor table, which records residual energy and IDs of all potential forwarders. The sender node concatenates IDs of a set of potential forwarders from the neighbor table in the packet. After receiving the packet, each potential forwarder searches for its ID in the packet. If a forwarder does not find its ID in the given set, the packet is discarded; however, if the ID is found, the residual energy is compared with that of the neighbors. The packet is forwarded immediately in case of maximum residual energy of the forwarder in its neighbors. The remaining forwarders will calculate the holding time and defer the transmission until the expiry of the timer. A node, while its timer is not expired yet and it receives a copy of the holding packet, will compute a random number. After the expiry of the timer, the random number will decide whether to forward the packet or not. The node discards the packet if the computed random number is greater than the PDR specified in the packet and multiple copies of the data packet are received. A node will forward a packet in the following two cases: either the computed random number is greater than the specified PDR, or it did not hear the transmission of the same packet. EEDBR successfully enhances the performance of DBR. However, it fails to successfully control duplicate packets, especially in the dense network scenario.

Authors in [42] proposed a depth-based routing scheme, which is an improved version of DBR, called Weighting Depth and Forwarding Area Division DBR (WDFAD-DBR). The forwarding strategy lies in the category of receiver-based routing. WDFAD-DBR considers the weighting depth difference of two hops; unlike DBR, which only considers one-hop depth difference in calculating the holding time. Additionally, the forwarding area is divided into three sections: a primary region, and two secondary regions for forwarding, so that high priority nodes can suppress low priority nodes. As it takes into account the depth of both the current forwarding neighbor and the expected next-hop neighbor, WDFAD-DBR attempts to avoid the void hole in advance. WDFAD-DBR successfully increases the PDR in a sparse network while reducing the energy usage and end-to-end delay. Lower depth nodes are penalized by the forwarding technique, which fails to balance network traffic, and thus causes the network to die early. It fails to reduce duplicate packets by stretching holding time differences among the neighboring nodes. Also, it fails to improve the performance in the dense network scenario.

Location-free Link State Routing (LLSR) is presented in [39]; a scheme which is receiver-based and takes forwarding decisions based on depth, link quality, and hop-count. Sink node initiates a beacon message with zero hop-count; the node on receiving the beacon message just increments the hop-count value and records it. After recording the hop-count value, it further broadcasts the beacon message with the updated value of the hop-count so as to reach the last node in the network. The depth and information about the hop count for each node is broadcasted. To make routing decisions, each node maintains a neighbor table containing one-hop neighbors' information. The node with the lowest hop-count, good quality path, and smaller depth will forward the packet. The residual energy of the node is ignored during routing due to which low energy nodes are penalized and they die quickly. Moreover, a large overhead is produced due to the beacon messages, resulting in the wastage of energy.

In [46], the authors proposed Reliable and Energy-efficient Opportunistic Routing, where the forwarding process selects optimal transmitting power and set of nodes to effectively postpone the death of the network. Additionally, two of the most critical problems, i.e., power conservation and reliable delivery of data are considered in designing the routing protocol. This results in prolonging the life of the network and data reliability. The experimental results of the proposed protocol demonstrated significant improvement, reliability, and reduced energy consumption. However, the proposed scheme failed to consider the energy difference among neighbors to reduce duplicate packets in unbalanced parts of the network.

Another receiver-based routing protocol called DOlphin Whale-Pods Routing (DOW-PR) is proposed in [43], in which the authors considered depth, hop-count, potential forwarding nodes, and suppressed nodes (nodes having higher depth than the sender) in calculating the holding time for a packet. The proposed scheme limits the transmission power of a node during forwarding. DOW-PR takes advantage of the suppressed nodes on encountering a void hole. In the absence of a potential forwarder, the packet is routed through the suppressed nodes to cope with the void hole. DOW-PR successfully reduced energy consumption using the following techniques: by taking nodes of lower hop-count towards the sink, by using controlled transmission power, and by selecting low traffic path in forwarding the packet. Moreover, it reduces packet loss by using suppressed nodes for forwarding in case of the void hole. However, DOW-PR penalizes low energy nodes due to not considering residual energy in computing the holding time.

The authors in [44] proposed a receiver-based routing protocol called Energy Balanced Efficient and Reliable Routing ($EBER^2$) to bring further improvement to WDFAD-DBR. It considers the weighting sum of depth difference of two hops, energy of the forwarding neighbor, as well as the receivers' neighbors, which can be the potential forwarders when calculating the holding time for a packet. $EBER^2$ allows the sender to set its transmission range (in other words, transmission power) based on the distance from the farthest node in the neighbor list. $EBER^2$ uses eleven sinks; nine sinks on the surface of water, and two sinks under the water. The two underwater sinks are deployed at those positions in water that have high node density. Large numbers of successful deliveries are performed by using the underwater sinks because a large percentage of network traffic lies in high node density areas. In addition, a significant reduction in energy consumption is observed. $EBER^2$ selects a high node density area for forwarding packets, which results in a high percentage improvement in the PDR. However, it fails to successfully achieve reduction in duplicate packets based on the average energy difference of the neighboring nodes, resulting in significant network energy wastage.

In [45] a receiver-based routing protocol is proposed, called Reliable Path Selection and Opportunistic Routing (RPSOR), which mainly focuses on reliability and packet advancement guarantee in UWSNs. To select a node, RPSOR calculates three variables: Advancement Factor (ADV_f), Reliability Index (REL_i), and Shortest Path Index (SP_i). ADV_f uses exponential function to find a weighting sum of the depth difference of two hops. The purpose of the exponential function is to create a large difference among neighboring nodes' holding time for a small difference in depth, in contrast to WDFAD-DBR, which uses a linear relationship between depth difference and holding time. REL_i is calculated by considering the average energy in the potential forwarding region to select a region having a high amount of energy but a small number of nodes. This strategy of calculating average energy results in significant improvement in the PDR. To select the shortest path towards the sink, RPSOR embeds SP_i in the holding time equation. SP_i is calculated based on hop-count and average depth difference among the nodes in the expected forwarding region.

Most of the depth and energy-based routing schemes, as discussed above, only consider depth and energy to calculate the holding time [47]. Some of the routing schemes take link quality by considering hop-count in the holding time equation. However, it requires exchanging many small packets for the current link quality due to which routing overhead is produced, which directly affects the lifetime of the node battery. To resolve this problem, in this paper, we propose a receiver-based routing protocol called Shifted Energy Efficiency and Priority (*SHEEP*) for UWSNs. In Section 3.3, we discuss how *SHEEP* computes the holding time and makes forwarding decisions.

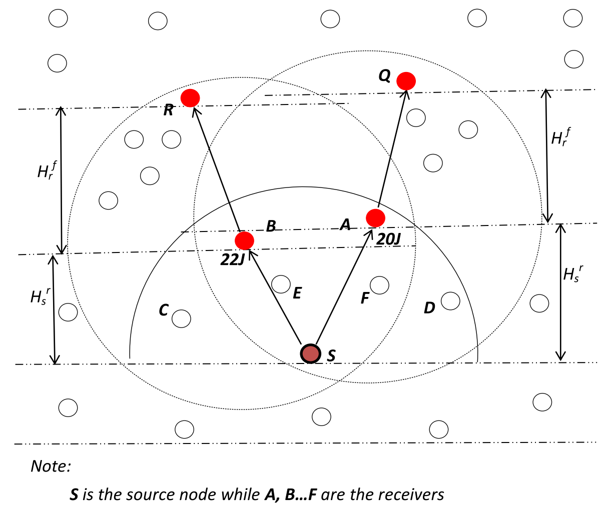

Fig. 2. Forwarder selection of $EBER^2$.

3. Proposed scheme

3.1. Problem formulation

The existing depth-based routing protocols fail in balancing network energy consumption. This is mainly because of the inappropriate design of the holding time mechanism or lack of some variables that are necessary to add the feature of energy balancing to the routing protocol. This section discusses the routing mechanism and shortcomings of an existing routing protocol called $EBER^2$ [44]. $EBER^2$ takes routing decisions based on depth of the current forwarder as well as depth of the expected forwarding node. It also considers the energy of the receiver and the potential forwarders of the receiver when calculating the holding time. The problem of energy and traffic balancing has been resolved to some extent; however, the approach of $EBER^2$ is far from perfect and cannot perform balancing successfully.

For example, Fig. 2 shows the routing mechanism of $EBER^2$ where a packet is broadcasted by the source node S and is received by all the potential forwarders, $A, B, \dots, F$. Let us assume A and B as competing forwarders. The weighting depth difference of node A is higher than that of B. However, according to the forwarding mechanism of $EBER^2$, node B qualifies to advance the communication of the packet. Because, in addition to the highest residual energy, node B also has larger number of potential forwarding nodes (PFNs).

The above discussion depicts the following shortcomings of $EBER^2$ routing protocol:

1. It only considers the value of residual energy of the current forwarding node in computing the fitness function value.
2. It always looks for a node having a large number of potential forwarders and furthers the communication through that node, which may result in large numbers of duplicate packets in the near future as no intelligent technique is available to reduce duplicate packets in dense parts of the network.
3. It fails to select a node having a small number of nodes and a large average difference among their residual energy.
4. It does not have any successful strategy to balance the unbalanced parts of the network.

The fitness function for $EBER^2$ is given in Eq. (1). By using the fitness function value, the node decides on forwarding the packet. Because, for large value of the fitness function, the holding time value comes out to be small and vice versa.

$$FF = \frac{\alpha H_s^r + (1-\alpha) H_r^f}{1 + Energy \times pfns} \tag{1}$$

Table 2
Nomenclature.

Symbol	Description
Tid	*ID* of the transmitter node
Rid	*ID* of the receiver node
ThE	Minimum energy which enables a node to participate in communication
PFNs	Potential forwarding nodes/neighbors of the receiver
PFNsID	Represents array of *IDs* of the *PFNs*
PFNsThE	PFNs having energy greater than *ThE*
pfns	Normalized value of *PFNs*
E_{max2}	Maximum energy in *PFNsThE*
E_{diff}	Differences of energy of the *PFNsThE* with respect to E_{max2}
CNGs	Common neighbors of the transmitter and receiver and having depth less than that of the transmitter
E_{max}	Maximum energy in *CNGs*
E_r	Residual energy
FF	Fitness function
T_r	Transmission range of a node
HT	Holding time
$V_{acoustic}$	Speed of the acoustic signal
HF	Healing factor
EL_f	Eligibility function
E_{tax}	Energy tax
E2ED	End-to-End delay
PDR	Packet delivery ratio

where,

$$H_s^r = senderdepth - receiverdepth$$

$$H_r^f = receiverdepth - forwarderdepth$$

$$Energy = 1 - \frac{E_r}{E_i}$$

$$pfns = 1 - \frac{PFNs}{NG_r}$$

where, E_r represents the residual energy of the receiver, E_i is the initial energy of the receiver, PFNs are the potential forwarding neighbors of the receiver, and NG_r is the total number of neighbors of the receiver. It means that the FF value is directly proportional to E_r and PFNs, because for the low value of Energy (alternatively high value of E_r) and PFNs (alternatively high value of PFNs), the denominator in Eq. (1) results in a higher number. The highest value FF node is the desired node to be selected for advancing the packet forwarding process; this is because of the short holding time, as shown in Eq. (2).

$$HT = \beta(T_r - FF) \tag{2}$$

where,

$$\beta = \frac{\frac{2T_r}{V_{acoustic}}}{\gamma}$$

Here, $\gamma \epsilon (0, T_r]$, where T_r is the predefined transmission range and $V_{acoustic}$ is the speed of the acoustic signal. In calculating the weighting sum of depth difference of the two hops, $EBER^2$ uses *alpha* to give weightage to the first and second hop neighboring nodes. The value of alpha remains constant throughout the lifetime of the network. $EBER^2$ fails to tune the value of alpha according to the information of the neighboring nodes. Due to the constant value of *alpha*, the forwarding strategy may select a node that is not appropriate for advancing the communication. This leads to the early death of the network, where a significant amount of energy still remains.

3.2. Proposed protocol

In this section, we describe our proposed *SHEEP* protocol and how it resolves the above-mentioned problems. To achieve balancing of energy consumption and reduction in duplicate packets, *SHEEP* embeds more features in designing the holding time, along with the depth difference of the current and expected forwarder. *SHEEP* does not use any constant in the forwarder selection scheme, in contrast to $EBER^2$, which uses alpha to give weightage to the current and expected next-hop forwarders. For the current hop, *SHEEP* prioritizes a node of the highest residual energy and lowest depth to balance energy consumption and traffic flow among the neighbors. This strategy removes the burden on a single node and avoids penalizing the highest depth nodes. A single node is not picked repeatedly due to which the death (means that the network is no more in a position to communicate) of the network is postponed. However, the highest residual energy strategy does not guarantee that the next forwarder will be picked up from a region where the average difference in residual energy of the neighbors is high. For this purpose, we embed another feature in designing the holding time called Healing Factor represented by *HF*, as shown in Eq. (3).

$$HF = \frac{\frac{\sum_{k=1}^{|NG_r|} E_{max2} - E_r^k}{|NG_r|}}{E_{max2}} \tag{3}$$

$$E_{max2} = Max(E_r^k | \forall k \epsilon NG_r)$$

$$E_r^k = residual\ energy\ of\ k th\ forwarder$$

$$E_{max2}, E_r^k \epsilon [0, E_i]$$

The *Healing Factor* makes the forwarding scheme more intelligent for selecting the next forwarder from a region where the average difference of the residual energy among the neighbors is higher. The following are the advantages of the *HF*:

- It always selects the next forwarder from a region where the average difference of the residual energy among the neighbors is higher and where the number of forwarding neighbors is low; in this way, packet collisions and duplicate packets are reduced
- It always looks for the unbalanced parts of the network, as discussed, where the average residual energy difference among the neighbors is high. This phenomenon may be called as *Treatment* of the unbalanced parts to make them balanced
- It does not pick nodes from a region which is already balanced
- It makes the holding time more intelligent to reduce void holes in advance by not allowing a packet through the path that results in packet loss
- It has the ability to distribute traffic along different paths of the network

Algorithm 1: Forwarder Selection of SHEEP

Input: $Node(m)\ received\ PKT\ from\ Node(n)$
$Tid = PKT.tid$
$Rid = Node(m).id$
Output: $NextforwarderID$
if $S(Tid).zd > S(Rid).zd$ ς ς $S(Rid).E > ThE$ **then**
  if $length(S(Rid).neighbors) > 0$ **then**
    $PFNs = find([S(S(Rid).neighbors).zd] > S(Rid).zd)$
    if $length(PFNs) > 0$ **then**
      $PFNsID = [S(Rid).neighbors(PFNs)]$
      $PFNsThE = find([S(PFNsID).E] > ThE)$
      if $length(PFNsThE) > 0$ **then**
        $E_{max2} = max([S([PFNsID(PFNsThE)]).E])$
        $E_{diff} = [E_{max2} - [S([PFNsID(PFNsThE)]).E]]$
        $HF = 1 - (((sum(E_{diff}))/(length(PFNsThE) - 1))/E_{max2})$
      $tempIDs = find([S(S(Rid).neighbors).zd] > S(Tid).zd)$
      $CNGs = (intersect([S(Rid).neighbors(tempIDs)], [S(Tid).neighbors]))$
      if $length(CNGs) > 0$ **then**
        if $length(find([S(CNGs).E] > 0)) > 0$ **then**
          $tempE = find([S(CNGs).E] > 0)$
          if $length(tempE) > 0$ **then**
            $E_{max} = max([S(CNGs(tempE)).E])$
  $EL_f = [((1 - (S(CNGs).E/E_{max})) * (1 - (S(Tid).zd - S(CNGs).zd)/T_r)) + (1 - S([PFNsID(PFNsThE)])/T_r) * HF]$
  $HT = [((2 * T_r)/v) * EL_f]$
  $NextforwarderID = S(min(HT)).id$

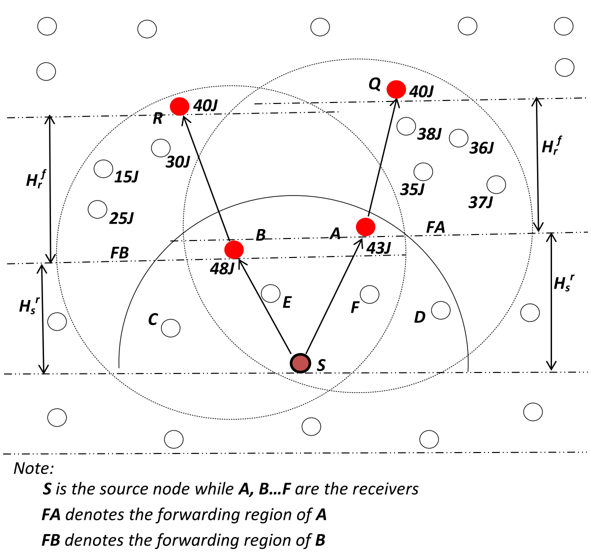

Fig. 3. Packet forwarding mechanism of SHEEP.

3.3. Holding time computation

In receiver-based routing, upon receiving a packet, the node holds it for some time based on certain conditions. This period is called the holding time during which the node waits for receiving a copy of the packet it is holding. In case no copy of the packet is received and the holding time expires, the node broadcasts the packet without any further delay. The details of the holding time computation of *SHEEP* are shown in Algorithm 1.

SHEEP uses the same strategy of calculating the holding time for every received packet, provided that the node receiving the packet is in the list of potential forwarders. Unlike opportunistic routing, SHEEP does not maintain an end-to-end route. In fact, the algorithm is based upon the local decisions at each receptor node. The memory is cleared once the forwarder is selected and the packet is transmitted.

Fig. 3 shows the strategy how *SHEEP* prioritizes the forwarder to advance the communication process. The source node S has two potential forwarders: A and B, which are approximately of the same depth. The holding time, in the case of *SHEEP*, depends on the value of the Eligibility Function, mentioned in Eq. (4), represented by EL_f; smaller the value of EL_f, smaller will be the holding time of the packet, and vice versa. EL_f decides whether the node is eligible or not to forward the packet. The value of EL_f depends on the receiver's depth, next hop forwarder's depth, residual energy of the receiver, and the average difference in the residual energy of the potential forwarding neighbors of the receiver (refer to Eq. (4)). According to the prioritizing scheme, node B is the selected forwarder for the received packet from S. This is due to the following reasons: firstly, *HF* value for B is higher because the average energy difference in the forwarding region F_B is higher as compared to F_A; although average energy is higher in F_A. Secondly, B has higher residual energy as compared to A. As a result, the value of EL_f comes out to be lower, which means that B is eligible to further forward the packet. On the other hand, A will suppress the packet transmission after receiving a copy from B.

$$HT = \frac{2T_r}{V_{acoustic}} \times EL_f \tag{4}$$

where,

$$EL_f = \left(1 - \frac{E_r}{E_{max}}\right) \times \left(1 - \frac{H_s^r}{T_r}\right) + \left(1 - \frac{H_r^f}{T_r}\right) \times (1 - HF)$$

E_r represents residual energy of the receiver, T_r represents transmission range of the receiver, and *HF* represents the Healing Factor of the receiver.

4. Simulation and results

In this section, we discuss the simulation setup and analysis of the results. We also compare the performance of SHEEP with that of $EBER^2$.

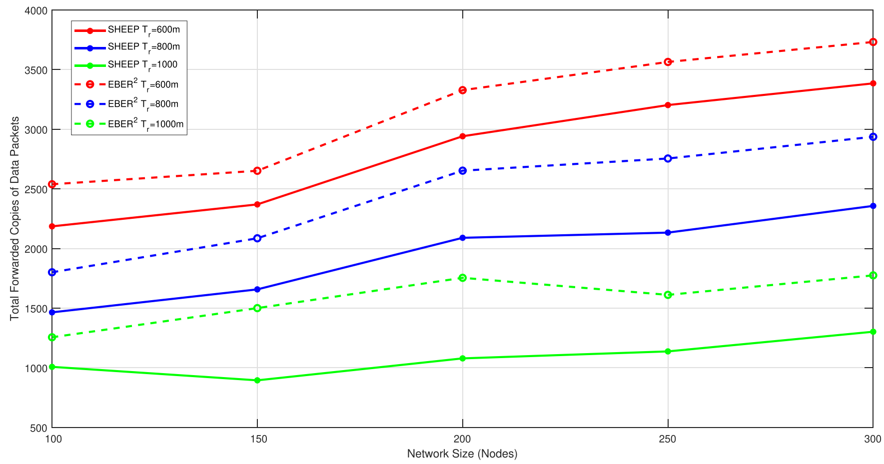

Fig. 4. The effect of network size on the total number of data copies forwarded.

4.1. Simulation setup

A three-dimensional area with $X = 1000$ m, $Y = 1000$ m, and $Z = 2000$ m is considered in our simulation setup. The network size of 100 nodes is considered initially, which is then incremented by 50 until 300 nodes to test the performance in a sparse as well as dense network scenario. After incrementing by 50, the newly deployed nodes are deployed randomly without affecting the previously deployed nodes. Similarly, the transmission range of 600 m to 1000 m is considered to see how the network performs for different transmission ranges. The deployment of nodes is random except the source/sink nodes as they are deployed at the bottom and surface of the sea, respectively. The network is considered to be homogeneous and each node is given an initial energy of 60 J.

A dynamic drift of nodes is considered such that the probability of moving in either direction (left or right) is 0.5, with a speed of 3 m/s. Only horizontal movement is considered because the nodes cannot move in the vertical direction due to high pressure of water.

The power of 5 W, 158 mW, and 50 mW is considered for sending, receiving, and ideal state, respectively. Three types of packets are used: data, neighbor request, and acknowledgment packets. Neighbor request and acknowledgment packets are of size 48 bits, while the data packet has a size of 664 bits, with 88 bits header and 576 bits payload size. The acoustic propagation speed of 1500 m/s and the data rate of 16 kbps is considered throughout the simulation of the network. The detailed results of the simulations are discussed in the following subsections:

4.2. Forwarded copies of data

Figs. 4 and 5 show the forwarded copies of the messages in SHEEP against $EBER^2$. As depicted in the figure, the increase in the network size increases the forwarded data for both the routing schemes. The reason is apparent, i.e., with the increase in the number of nodes, the chances of duplicate packets increase accordingly. In a dense network scenario, the nodes are close enough to one another due to which some close neighbors have the smaller holding time difference than the propagation distance among them.

The duplicate packet may be transmitted due to many reasons, e.g., when the holding time of the packet finishes before a node can hear the transmission from some high priority node.

A node may transmit a packet and the highest priority node does not receive it yet. Poor design of the holding time equation is another reason for duplicate packets. However, the trend is opposite in case of varying the transmission range, as shown in Fig. 5. As depicted in the figure, when the number of nodes is constant, the forwarded copies of packets decrease with the increase in transmission range for both SHEEP and $EBER^2$. This is because for the higher transmission range, the node will cover a large area and high packet suppression is achieved. It is also observed that SHEEP has lower forwarded copies compared to $EBER^2$. This happens mainly due to two reasons; (i) the design of the fitness function equation in $EBER^2$ cannot successfully increase the holding time difference among the neighbors because it simply takes residual energy of the receiver and places it in the fitness function equation. There is no information available about the difference of the residual energy among neighbors on the second hop, due to which the routing scheme is unable to pick the forwarder from a region where the difference in the residual energy among neighbors is high, as done in SHEEP. The advantage of selecting the forwarder from a high difference residual energy region is that the holding difference will be high among the neighbors and the chances of duplicate packets will be small. (ii) $EBER^2$ looks for a dense region to pick a forwarder from; however, no intelligent technique is available to reduce duplicate packets in that region in contrast to SHEEP, which selects the forwarder from a region having a small number of nodes but with high average difference among their residual energy.

4.3. Energy consumption and energy tax

Energy Consumption refers to the sum of the transmission energy, receiving energy, ideal state energy, and processing energy, while Energy Tax is the total energy consumption per node on the successful delivery of a packet from source node to the destination node, as calculated in Eq. (5). It is represented by E_{tax}. Mathematically,

$$E_{tax} = \frac{Total\ Energy\ Consumption}{Nodes \times Packets} \tag{5}$$

Energy Consumption: Energy consumption comparison of SHEEP with $EBER^2$ is shown in Figs. 6 and 7 on varying the node number and transmission range, respectively. It is observed that the energy consumption of the network increases as the simulation time progresses for both SHEEP and $EBER^2$. The graph becomes straight at the end of the simulation time because most of the nodes deplete at this stage and there is no energy left to consume. The energy consumption increases when the network size is increased and vice versa. This is because of the forwarding copies of data, as discussed in Section 4.2. However, no significant change is observed in the case of varying transmission ranges.

SHEEP consumes lower energy compared to $EBER^2$ on every point of the simulation time. This is because of the energy balancing technique used in SHEEP that prioritizes a forwarder, which can result in maintaining the balanced energy consumption throughout the network life. The percentage improvement in energy consumption increases with the network size because of the fact that $EBER^2$ always selects the

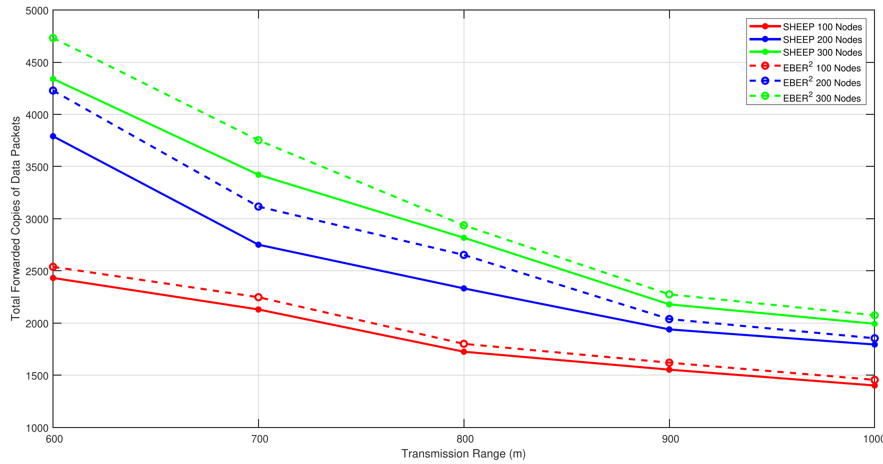

Fig. 5. The effect of transmission range on the total number of data copies forwarded.

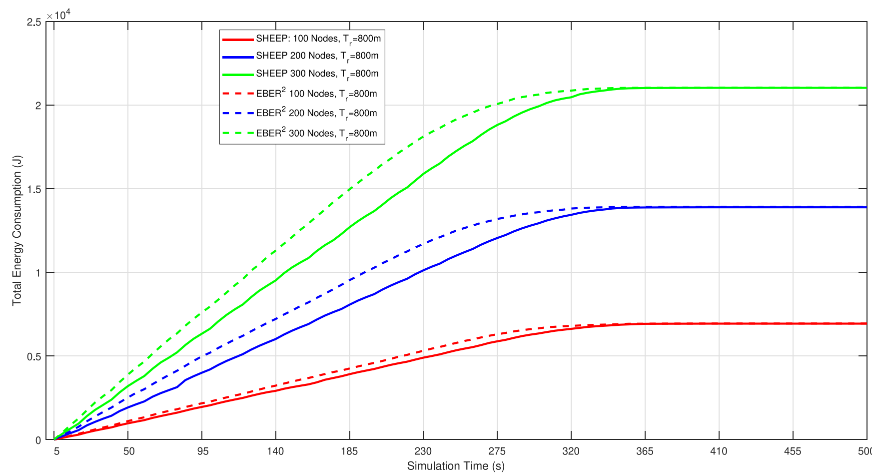

Fig. 6. The effect of network size on energy consumption.

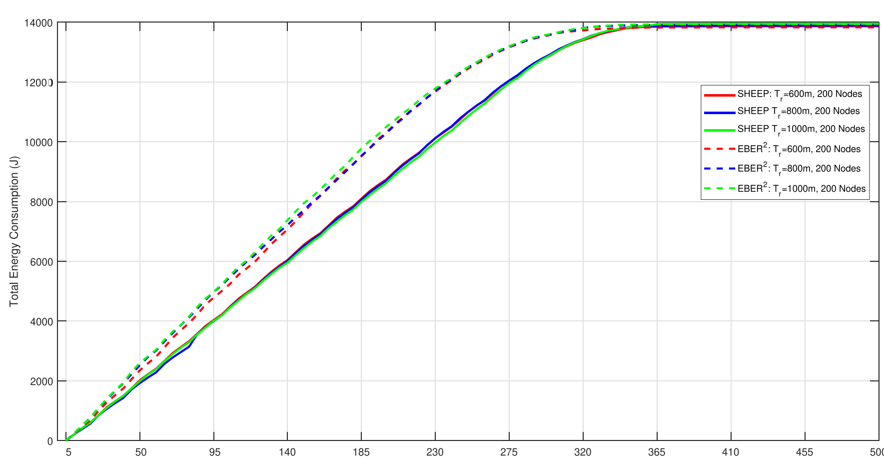

Fig. 7. The effect of transmission range on energy consumption.

node having many potential forwarding nodes. Therefore, in a denser network, the chances of duplicate packets will be high, in contrast to SHEEP, which selects the forwarder from a region where the transmitter node can suppress its neighbors by a large difference in the holding time. In addition, SHEEP consumes relatively less energy due to the reduced number of duplicate packets. This point is also discussed in Section 4.2 in detail. However, the behavior of both the routing schemes is almost the same for 100 nodes because $EBER^2$ can perform great in a sparse network scenario (see Table 3).

Energy Tax: The comparison of SHEEP against $EBER^2$ is shown in Figs. 8 and 9. It is observed that the E_{tax} of SHEEP is less than that of $EBER^2$ in case of sparse and dense networks as well as for small and large transmission ranges. This is because SHEEP carries out much more successful deliveries on a relatively small energy consumption as

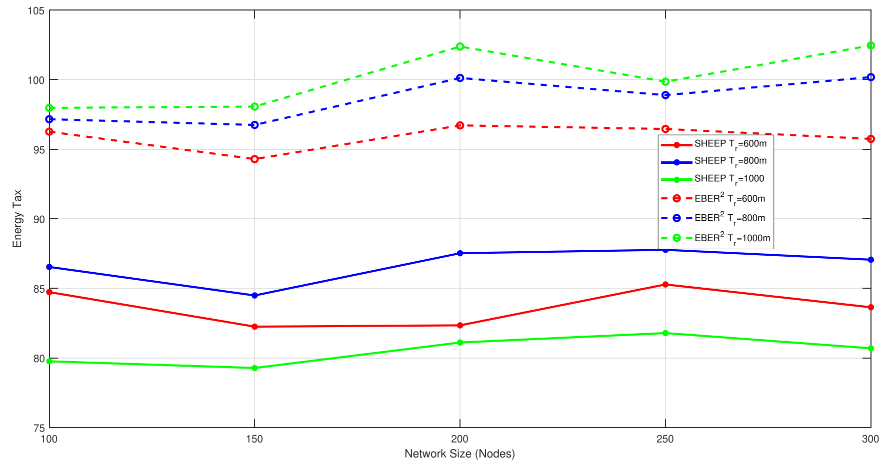

Fig. 8. Energy Tax with varying network size.

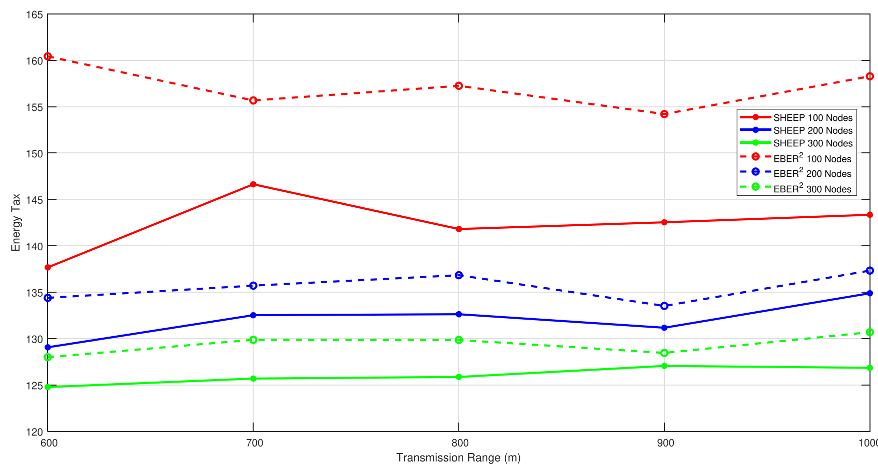

Fig. 9. Energy Tax with varying transmission range.

Table 3
Energy Consumption (%) improvement of SHEEP in comparison to $EBER^2$.

T_r	100 nodes	150 nodes	200 nodes	250 nodes	300 nodes
600 m	3.1%	3.0%	5.7%	5.8%	7.2%
800 m	2.9%	2.8%	4.2%	4.8%	5.1%
1000 m	5.7%	7.1%	9.4%	7.1%	10%
Avg-Imp	3.9%	4.3%	6.4%	5.9%	7.4%

Table 4
Energy Tax (%) improvement of SHEEP in comparison to $EBER^2$.

T_r	100 nodes	150 nodes	200 nodes	250 nodes	300 nodes
600 m	10.6%	11.4%	9.5%	11.5%	11.7%
800 m	9.4%	11.4%	12.3%	10.4%	11.3%
1000 m	17.1%	17.4%	20%	17.2%	20.9%
Avg-Imp	12.3%	13.4%	13.9%	13.0%	14.6%

compared to the counterpart $EBER^2$, which consumes a large amount of energy on relatively small number of successful deliveries. Also, the E_{tax} increases with increase in the network size for a given transmission range. This is due to the increasing energy consumption, or alternatively, the increased number of forwarded data copies, resulting in higher energy consumption (see Table 4).

4.4. Packet delivery ratio

PDR is the ratio of the total number of successful packets delivered to the total number of packets generated at the source node, as given in Eq. (6). It is represented mathematically as follows:

$$PDR = \frac{total\ number\ of\ successful\ packets}{total\ number\ of\ generated\ packets} \tag{6}$$

In this part, we discuss PDR to see how the network performs in terms of data delivery when SHEEP is used as a routing scheme.

Figs. 10 and 11 show the PDR comparison of SHEEP with $EBER^2$ on varying number of nodes and transmission range, respectively. It is observed that SHEEP has a higher PDR than that of $EBER^2$ for varying node numbers or transmission range. The reasons are as follows: Firstly, the forwarded copies of data are higher in case of $EBER^2$, as discussed in Section 4.2, due to which a higher amount of network energy is consumed as compared to SHEEP, and hence the network dies in a relatively smaller interval of time and is unable to perform data delivery anymore. Figs. 16 and 17 show that the average operational time for the proposed scheme is higher than that of $EBER^2$. In addition, a higher number of dead nodes are created in the case of $EBER^2$, which is also a reason for the early death of the network. Secondly, due to a higher amount of forwarded copies of data, the chances of packet collision are higher for $EBER^2$ and a significant number of packet loss occurs, which results in decreased PDR. However, PDR is slightly decreased in the dense network scenario for both SHEEP and $EBER^2$. The reason is apparent; the forwarded copies of data increase in the

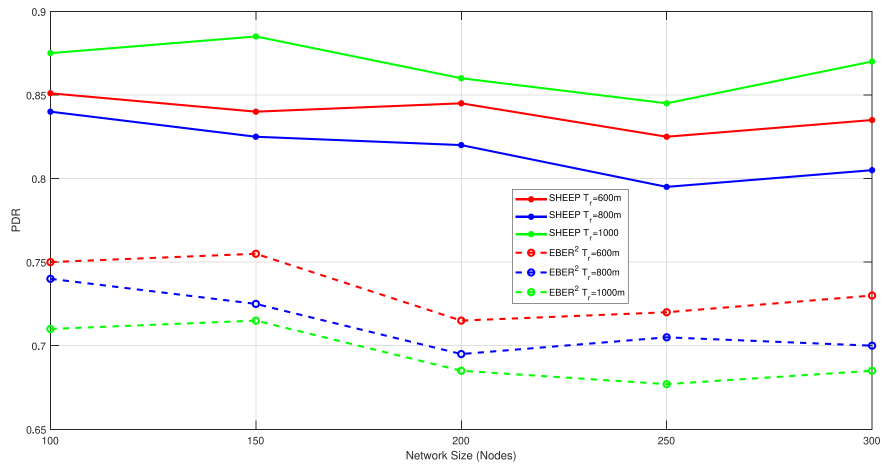

Fig. 10. The effect of network size on packet delivery ratio.

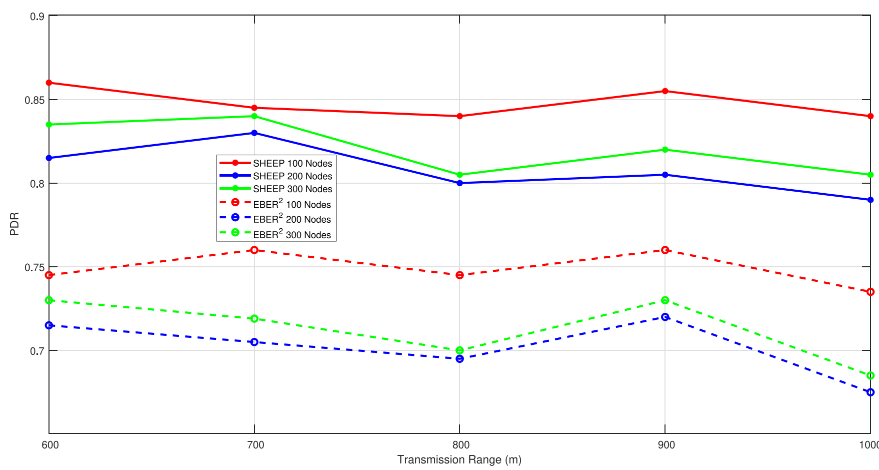

Fig. 11. The effect of transmission range on packet delivery ratio.

Table 5
PDR (%) improvement of SHEEP in comparison to $EBER^2$.

T_r	100 nodes	150 nodes	200 nodes	250 nodes	300 nodes
600 (m)	11%	10%	15%	13%	11%
800 (m)	10%	10%	13%	10%	10%
1000 (m)	16%	16%	12%	17%	18%
Avg-Imp	12.3%	12%	13.3%	13.3%	13%

dense network, resulting in high collision and packet losses, eventually leading to decreased PDR (see Table 5).

4.5. Dead nodes

If the node energy falls below the *ThE* (refer to Table 2) and cannot perform transmission anymore, the node is said to be dead. A void hole is created where a sensor node dies. Due to void holes, a large number of packet losses occur, or the packets are unable to move further in the network. Dead nodes are also a significant part of the simulation.

Fig. 14 shows the average number of dead nodes when the network size varies but transmission range is kept fixed. It is observed that the dead nodes increase with the network size for a single transmission range. This is because, with the higher network size, the number of participating nodes increases in order to continue the forwarding process. SHEEP has fewer dead nodes compared to $EBER^2$ for all sizes of the network. This is because of the holding time calculation approach of SHEEP that enables the routing scheme to successfully balance the usage of nodes for the forwarding process. This results in a large number of nodes that remain alive as compared to $EBER^2$.

Fig. 15 shows the average number of dead nodes with varying transmission range for fixed network size. The increasing transmission range does not have a significant effect on the number of dead nodes because the network size remains the same. However, with the increase in network size, the number of dead nodes increases accordingly, as discussed above.

Figs. 12 and 13 show the network performance in terms of dead nodes as the simulation time progresses. It is observed that the nodes start to deplete earlier in the case of $EBER^2$ as compared to SHEEP. As discussed in Section 4.3, a higher amount of energy is consumed in case of $EBER^2$ due to the higher number of forwarded copies of data packets, which is the reason why nodes start to die earlier as compared to SHEEP. This is due to the effective balancing technique embedded in the holding time calculation of SHEEP, which always selects the forwarder from a region of unbalanced residual energy to make it balanced by the use of healing factor. As a result, the network remains alive for a longer period of time and performs with higher number of deliveries, as discussed earlier in Section 4.4. The healing factor greatly contributes to the prolongation of the network lifetime. In contrast, $EBER^2$ just selects a forwarder of higher residual energy and fails to consider the difference of energy among the neighbors.

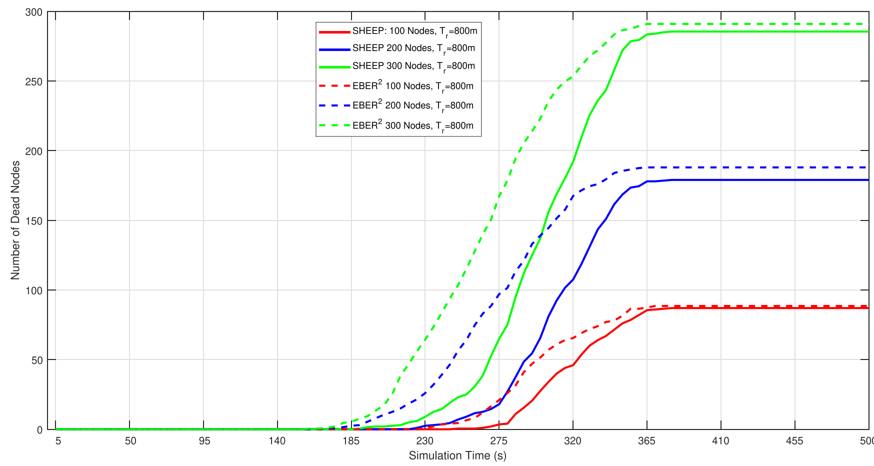

Fig. 12. The effect of simulation time on dead nodes with varying number of nodes.

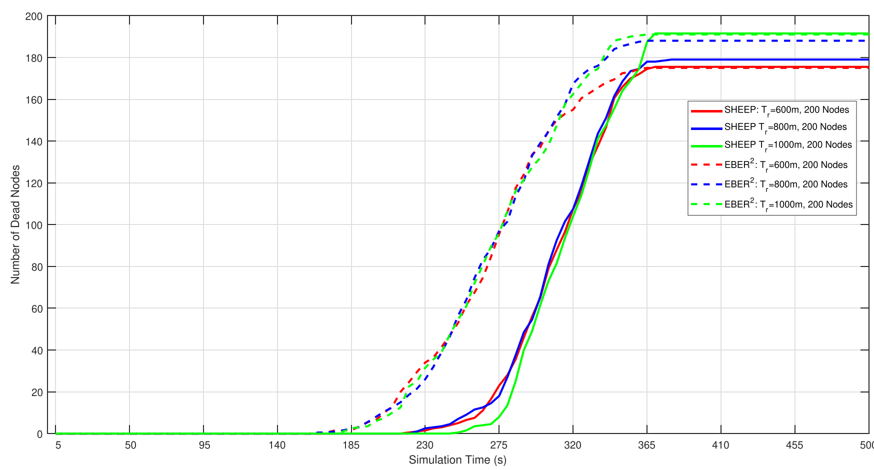

Fig. 13. The effect of simulation time on dead nodes with varying transmission range.

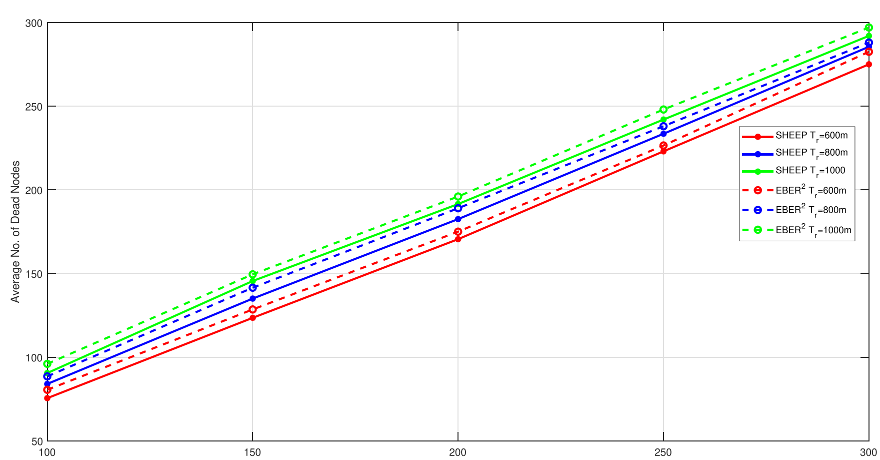

Fig. 14. The effect of network size on dead nodes.

4.6. Average operational time

Figs. 16 and 17 show the average operational time of SHEEP and $EBER^2$ for different network sizes and transmission ranges, respectively. Operational time is defined as the time passed to receive the last data packet by any of the sink nodes. It depends on the network lifetime, i.e., the operational time increases with the lifetime of the network and vice versa. It is observed that the average operational time of SHEEP is longer than $EBER^2$. This is because of the high energy consumption in case of $EBER^2$. The high energy consumption is the result of more forwarded packets, as discussed in Section 4.3. Such higher consumption of energy forces the nodes to deplete earlier as

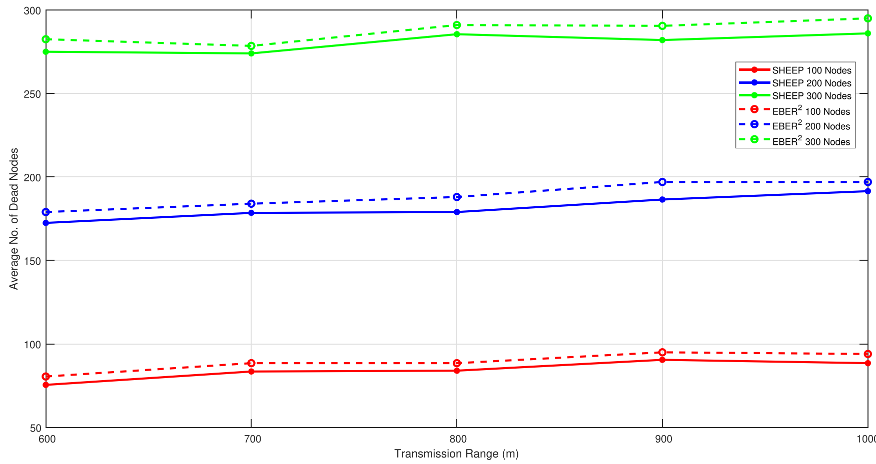

Fig. 15. The effect of transmission range on dead nodes.

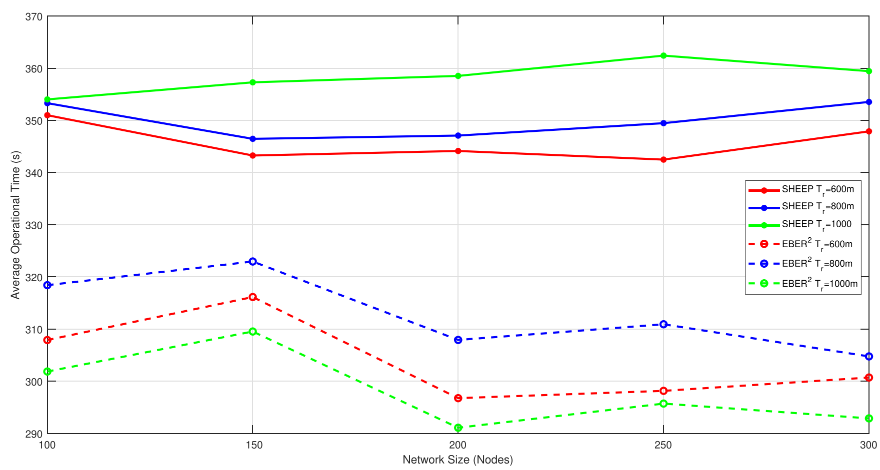

Fig. 16. The effect of varying network size on operational time.

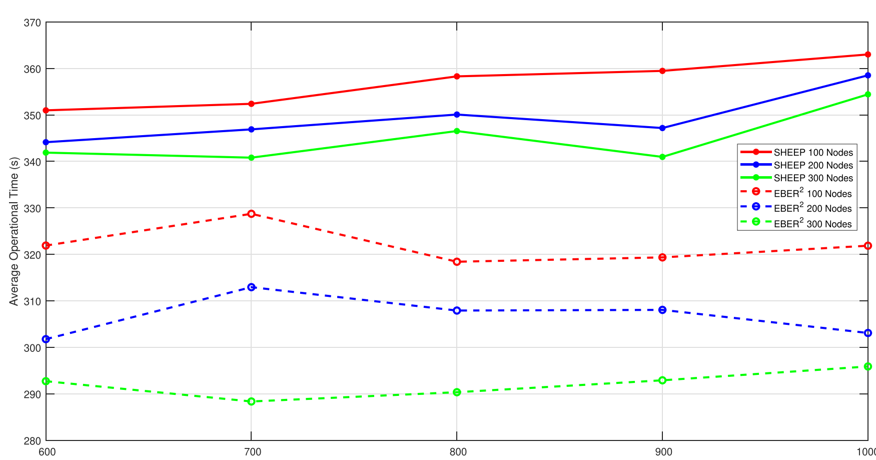

Fig. 17. The effect of varying transmission range on operational time.

compared to SHEEP and the network cannot survive for a longer period of time, as discussed in Section 4.5.

Fig. 16 shows that the increase in the network size has no significant effect on the operational time of the network in case of SHEEP. This is because of the energy balancing strategy, which has the capability of balancing the unbalanced parts of the network, even in a dense network scenario, by embedding the healing factor in the Eligibility Function. On the other hand, the average operational time in case of $EBER^2$ decreases with an increase in the network size. This is because $EBER^2$ always selects a node from large number of potential forwarding nodes in its transmission range, due to which duplicate packet generation rate increases. This results in higher energy consumption and earlier death of the network. However, the average operational time of SHEEP has slightly increased with an increase in the transmission range of the

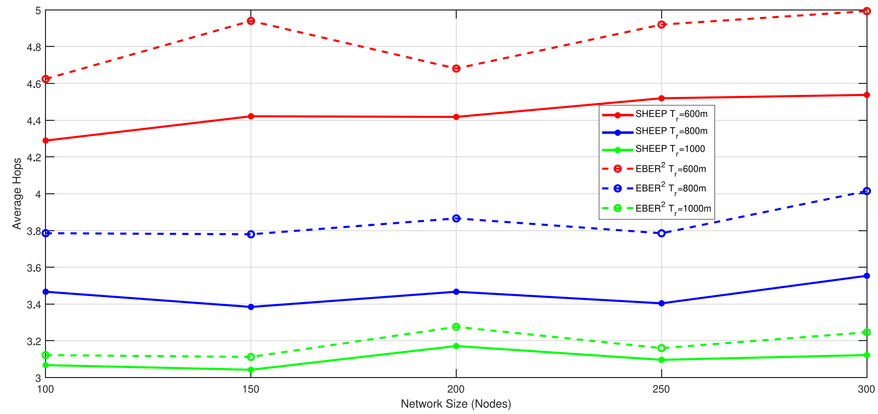

Fig. 18. Number of hops with varying network size.

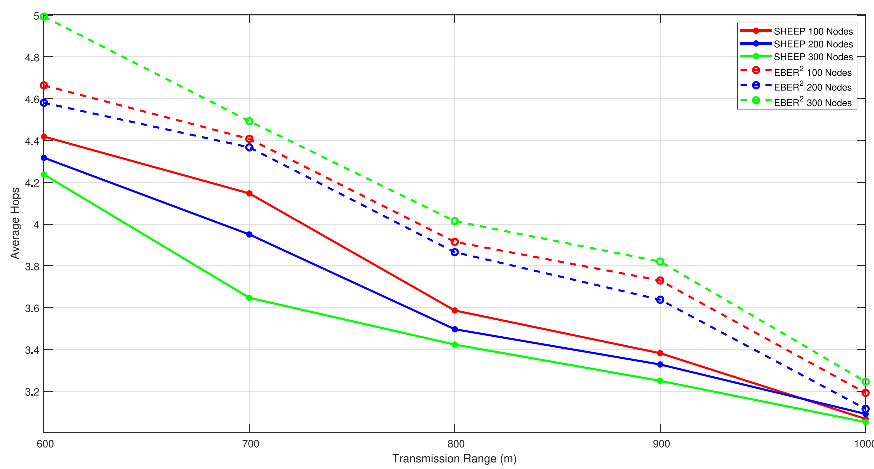

Fig. 19. Number of hops with varying transmission range.

nodes. All the results described in this section demonstrate the stability of the network for varying node density and transmission range for SHEEP in UWSNs.

4.7. End to End delay (E2ED)

It is the time a packet needs to reach from the source to the destination node or sink. It includes transmission time, receiving time, and propagation delay. This subsection consists of a detailed discussion about the comparison of the E2ED for both the routing schemes. Figs. 20 and 21 show the E2ED of SHEEP and $EBER^2$ for varying number of nodes and transmission range, respectively. The E2ED of the proposed scheme is lesser than $EBER^2$ for various network sizes as well as transmission ranges. This is because of the higher energy consumption in case of $EBER^2$. It forwards the packets through denser paths of the network, which causes higher unwanted packet broadcasting due to which some parts of the network die earlier. In order to continue the communication, the packets have to follow some other longer path towards the sink. In this way, the number of hops increases to reach the destination or sink node (see Table 6).

Figs. 18 and 19 depict the number of hops for the two protocols (SHEEP and $EBER^2$) on varying network size and transmission range, respectively. SHEEP has lower number of hops compared to $EBER^2$. The balancing technique contributed greatly to reduce the delay in SHEEP because the network stays alive for a longer period of time and the shortest path remains till the end of the network. The communication does not need to continue through the longer path as in case of $EBER^2$. This is because all parts of the network almost die at the same point in time. The lesser number of hops in case of SHEEP also contributed to the decrease in energy consumption.

Table 6
E2ED (%) improvement of SHEEP in comparison to $EBER^2$.

T_r	100 nodes	150 nodes	200 nodes	250 nodes	300 nodes
600 m	10%	10%	8.8%	10.5%	15.7%
800 m	12.7%	18.7%	12.3%	18.5%	21.8%
1000 m	15.6%	15.7%	12.5%	12.8%	18.8%
Avg-Imp	12.7%	14.8%	11.2%	13.9%	18.7%

4.8. Control overhead

Control overhead or routing overhead occurs due to the redundant packets (acknowledgment and neighbor request packets in case of SHEEP). By exchanging these small packets, nodes can keep updated information about their neighbors. These packets are generated once for a complete round of the simulation. To find the control overhead of SHEEP, we compare the processing time of the redundant packets (acknowledgment and neighbor request) and data packets. Fig. 22 shows the number of redundant packets in comparison to data packets, while Fig. 23 shows the total duration occupied by the redundant and data packets. It can be observed that only a small percentage of the processing time is occupied by the redundant packets because the size of the redundant packets is very small as compared to the data packets.

5. Conclusion and future work

This paper presented a novel routing protocol called Shifted Energy Efficiency and Priority (SHEEP) for Underwater Wireless Sensor

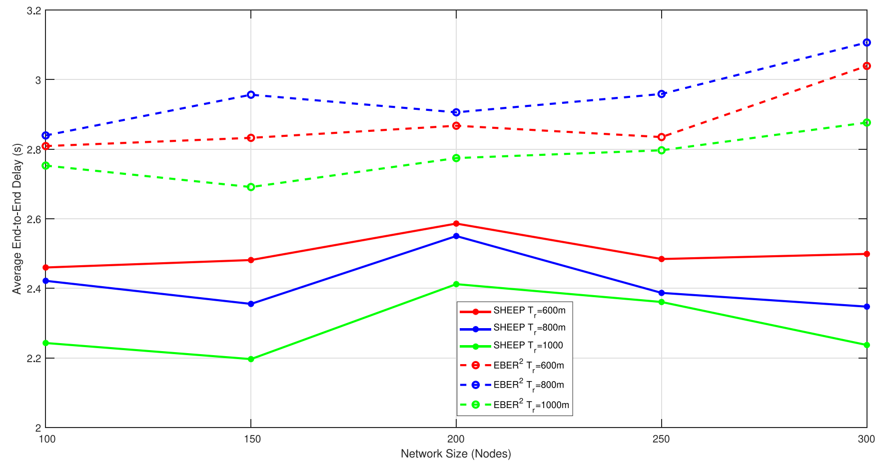

Fig. 20. The effect of varying network size on end-to-end delay.

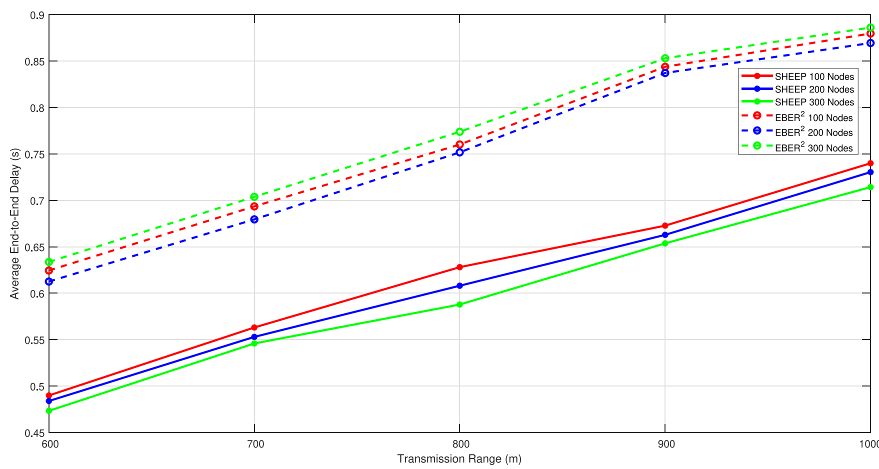

Fig. 21. The effect of transmission range on end-to-end delay.

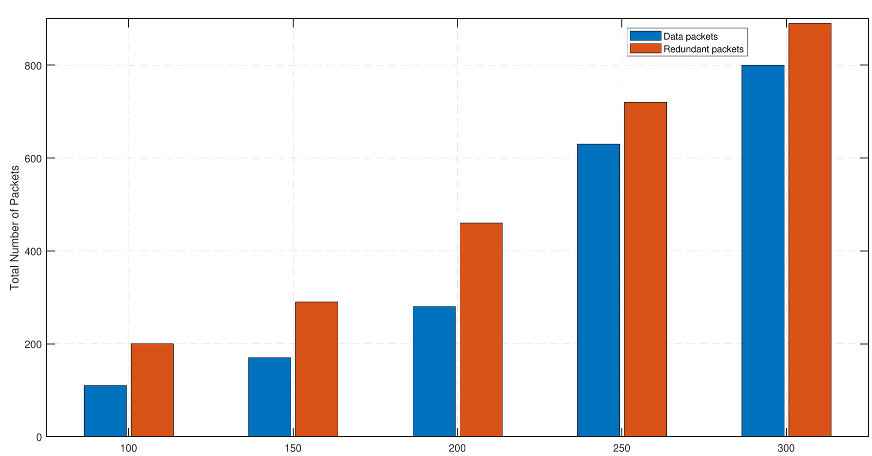

Fig. 22. Total number of generated packets in SHEEP protocol.

Networks (UWSNs). The architecture of SHEEP is based on the state-of-the-art Energy Balanced Efficient and Reliable Routing ($EBER^2$) protocol as it positively shifts its energy efficiency level and other performance metrics while focusing more on energy efficiency. It prioritizes a node based on the residual energy and average energy difference among the expected forwarders, called the Healing Factor. Extensive simulations of our proposed protocol were performed in various scenarios and the results obtained show that SHEEP has successfully reduced the forwarded copies of data, network energy consumption, End-to-End-Delay (E2ED), and E_{tax}. In addition, it also increased the Packet

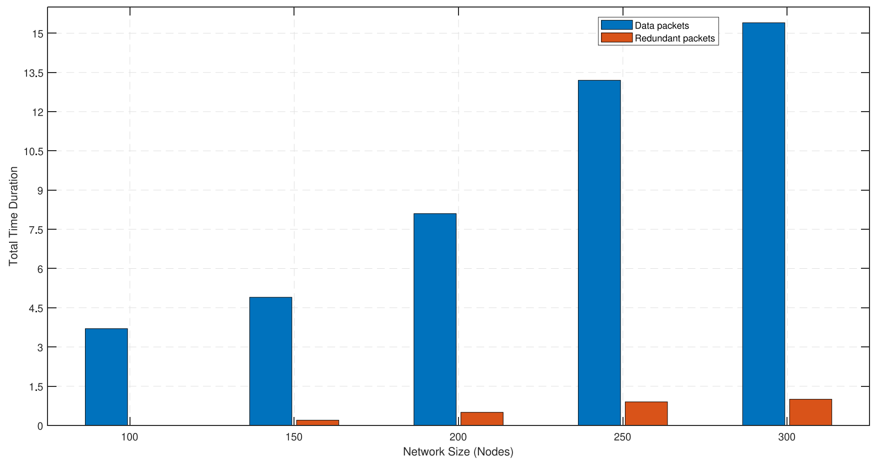

Fig. 23. Total duration occupied by redundant and data packets.

Delivery Ratio (PDR) and the network lifetime. We have also performed the quantitative analysis, which shows a significant improvement in network performance when SHEEP is used as a routing protocol. However, the forwarded copies of data are more in the dense network (though these are still lower than those produced by $EBER^2$) due to which the average operational time of the network is reduced.

In future, the performance of SHEEP can be improved by embedding these important features in the routing scheme to make the operational time stable irrespective of the size of the network.

Declaration of competing interest

The authors declare that they have no known competing financial or personal interests that could have appeared to influence the work reported in this paper.

References

[1] J. Li, J. Wu, C. Li, A.K. Bashir, J. Li, Y.D. Al-Otaibi, Information-centric wireless sensor networking scheme with water-depth-awareness content caching for underwater IoT, IEEE Internet Things J. (2021).
[2] M.A. Basset, R. Mohamed, M. Elhoseny, A.K. Bashir, A. Jolfaei, N. Kumar, Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications, IEEE Trans. Ind. Inform. 17 (7) (2021) 5068–5076.
[3] S. Khan, A.N. Alvi, M.A. Javed, Y.D. Al-Otaibi, A.K. Bashir, An efficient medium access control protocol for RF energy harvesting based IoT devices, Comput. Commun. 171 (2021) 28–38.
[4] E. Felemban, F.K. Shaikh, U.M. Qureshi, A.A. Sheikh, S.B. Qaisar, Underwater sensor network applications: A comprehensive survey, Int. J. Distrib. Sens. Netw. 11 (11) (2015).
[5] K. Alam, T. Ray, S.G. Anavatti, Design optimization of an unmanned underwater vehicle using low- and high-fidelity models, IEEE Trans. Syst. Man, Cybern. Syst. 47 (11) (2017) 2794–2808.
[6] T. Xu, J. Yu, C. Vong, B. Wang, X. Wu, L. Zhang, Dynamic morphology and swimming properties of rotating miniature swimmers with soft tails, IEEE/ASME Trans. Mech. 24 (3) (2019) 924–934.
[7] P.R. Bandyopadhyay, Trends in biorobotic autonomous undersea vehicles, IEEE J. Ocean. Eng. 30 (1) (2005) 109–139.
[8] D.T. Roper, S. Sharma, R. Sutton, P. Culverhouse, A review of developments towards biologically inspired propulsion systems for autonomous underwater vehicles, Proc. Inst. Mech. Eng. M J. Eng. Mar. Environ. 225 (2) (2011) 77–96.
[9] W. Chu, et al., Review of biomimetic underwater robots using smart actuators, Int. J. Precis. Eng. Manuf. 13 (7) (2012) 1281–1292.
[10] A. Raj, A. Thakur, Fish-inspired robots: Design, sensing, actuation, and autonomy—A review of research, Bioinspir. Biomi. 11 (3) (2016) 031001.
[11] D. Scaradozzi, G. Palmieri, D. Costa, A. Pinelli, BCF swimming locomotion for autonomous underwater robots: A review and a novel solution to improve control and efficiency, Ocean Eng. 130 (2017) 437–453.
[12] J. Yu, L. Wen, Z. Ren, A survey on fabrication, control, and hydrodynamic function of biomimetic robotic fish, sci, China Technol. Sci. 60 (9) (2017) 1365–1380.
[13] J. Yu, M. Wang, H. Dong, Y. Zhang, Z. Wu, Motion control and motion coordination of bionic robotic fish: A review, J. Bionic Eng. 15 (4) (2018) 579–598.
[14] N. Tariq, F.A. Khan, M. Asim, Security challenges and requirements for smart internet of things applications: A comprehensive analysis, in: The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2021), Leuven, Belgium, 2021, pp. 9–12.
[15] A. Ali, F.A. Khan, A broadcast-based key agreement scheme using set reconciliation for Wireless Body Area networks, J. Med. Syst. (Springer) 38 (5) (2014).
[16] R. Rizwan, F.A. Khan, H. Abbas, S.H. Chauhdary, Anomaly detection in wireless sensor networks using immune-based bio-inspired mechanism, Int. J. Distrib. Sens. Netw. 2015 (2015) 684952.
[17] D. Farashaei, A. Honarbakhsh, S.M. Movahedifar, et al., Individual flexibility and workplace conflict: cloud-based data collection and fusion of neural networks, Wirel. Netw. (2022) http://dx.doi.org/10.1007/s11276-022-02887-0.
[18] K. Yu, L. Tan, S. Mumtaz, S. Al-Rubaye, A. Al-Dulaimi, A.K. Bashir, F.A. Khan, Securing critical infrastructures: Deep learning-based threat detection in the iIoT, IEEE Commun. Mag. 59 (10) (2021) 76–82.
[19] A. Hazra, M. Adhikari, T. Amgoth, S.N. Srirama, Intelligent service deployment policy for next-generation industrial edge networks, IEEE Trans. Netw. Sci. Eng. (2021) http://dx.doi.org/10.1109/TNSE.2021.3122178.
[20] A. Hazra, M. Adhikari, T. Amgoth, S.N. Srirama, Collaborative AI-enabled intelligent partial service provisioning in green industrial fog networks, IEEE Internet Things J. (2021) http://dx.doi.org/10.1109/JIOT.2021.3110910.
[21] M. Adhikari, A. Munusamy, A. Hazra, V.G. Menon, V. Anavangot, D. Puthal, Security and privacy in edge-centric intelligent internet of vehicles: Issues and remedies, IEEE Consum. Electron. Mag. (2021) http://dx.doi.org/10.1109/MCE.2021.3116415.
[22] A. Hazra, P. Choudhary, O. Vivek, An advance mobility management scheme in wireless network, in: 2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT, 2018, pp. 1–5, http://dx.doi.org/10.1109/ICCCNT.2018.8493854.
[23] E.M. Sozer, M. Stojanovic, J.G. Proakis, Underwater acoustic networks, IEEE J. Ocean. Eng. 25 (2000) 72–83.
[24] M. Stojanovic, J. Preisig, Underwater acoustic communication channels: Propagation models and statistical characterization, IEEE Commun. Mag. 47 (2009) 84–89.
[25] X. Che, I. Wells, G. Dickers, P. Kear, X. Gong, Re-evaluation of RF electromagnetic communication in underwater sensor networks, IEEE Commun. Mag. 48 (2010) 143–151.
[26] U.M. Cella, R. Johnstone, N. Shuley, Electromagnetic wave wireless communication in shallow water coastal environment: Theoretical analysis and experimental results, in: Proceedings of the Fourth ACM International Workshop on Underwater Networks, vol. 3, Berkeley, CA, USA, 2009.
[27] H. Kaushal, G. Kaddoum, Underwater optical wireless communication, IEEE Access 4 (2016) 1518–1547.
[28] F. Hanson, S. Radic, High bandwidth underwater optical communication, Appl. Opt. 47 (2008) 277–283.
[29] J. Heidemann, W. Ye, J. Wills, A. Syed, Y. Li, Research challenges and applications for underwater sensor networking, in: Proceedings of the IEEEWireless Communications and Networking Conference, Las Vegas, NV, USA, 2006, pp. 3–6.

[30] I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: Research challenges, Ad Hoc Netw. 3 (2005) 257–279.

[31] J.E. Garcia, Accurate positioning for underwater acoustic networks, in: Proceedings of the 2005-Europe Oceans, Brest, France, 2005, pp. 20–23.

[32] P. Xie, J.-H. Cui, L. Lao, VBF: Vector-based forwarding protocol for underwater sensor networks, in: Proceedings of the 5th International IFIP-TC6 Conference on Networking Technologies, Services, and Protocols, Performance of Computer and Communication Networks, Mobile and Wireless Communications Systems, Coimbra, Portugal, 2006, pp. 15–19.

[33] N. Nicolaou, A. See, P. Xie, J.H. Cui, D. Maggiorini, Improving the robustness of location-based routing for underwater sensor networks, in: Proceedings of the 2007-Europe Oceans, Aberdeen, UK, 2007, pp. 18–21.

[34] D. Hwang, D. Kim, DFR: Directional flooding-based routing protocol for underwater sensor networks, in: Proceedings of the OCEANS 2008, Quebec City, QC, Canada, 2008, pp. 15–18.

[35] T. Ali, L.T. Jung, S. Ameer, Flooding control by using angle based cone for UWSNs, in: Proceedings of the 2012 International Symposium on Telecommunication Technologies, Kuala Lumpur, Malaysia, 2012, pp. 26–28.

[36] H. Yu, N. Yao, J. Liu, An adaptive routing protocol in underwater sparse acoustic sensor networks, Ad Hoc Netw. 34 (2015) 121–143.

[37] H. Yan, Z.J. Shi, J.-H. Cui, DBR: Depth-based routing for underwater sensor networks, in: Proceedings of the 7th International IFIP-TC6 Networking Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, Singapore, 2008, pp. 5–9.

[38] A. Wahid, S. Lee, H.-J. Jeong, D. Kim, Eedbr: Energy-efficient depth-based routing protocol for underwater wireless sensor networks, in: Advanced Computer Science and Information Technology;, Springer Verlag, Berlin/Heidelberg, Germany, 2011, pp. 223–234.

[39] M. Barbeau, S. Blouin, G. Cervera, J. Garcia-Alfaro, E. Kranakis, Location-free link state routing for underwater acoustic sensor networks, in: Proceedings of the 28th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE, Halifax, NS, Canada, 2015, pp. 3–6.

[40] S.M. Ghoreyshi, A. Shahrabi, T. Boutaleb, An inherently void avoidance routing protocol for underwater sensor networks, in: Proceedings of the Twelfth IEEE International Symposium on Wireless Communication Systems, Brussels, Belgium, 2015, pp. 25–28.

[41] S.M. Ghoreyshi, A. Shahrabi, T. Boutaleb, A novel cooperative opportunistic routing scheme for underwater sensor networks, Sensors 16 (2016) 297.

[42] H. Yu, N. Yao, T. Wang, G. Li, Z. Gao, G. Tan, WDFAD-DBR: Weighting depth and forwarding area division DBR routing protocol for uasns, Ad Hoc Netw. 37 (2016) 256–282.

[43] Z. Wadud, K. Ullah, S. Hussain, X. Yang, A.B. Qazi, DOW-PR dolphin and whale pods routing protocol for underwater wireless sensor networks (UWSNs), Sensors 18 (5) (2018) 1529.

[44] Z. Wadud, M. Ismail, A.B. Qazi, F.A. Khan, A. Derhab, I. Ahmad, A.M. Ahmad, An energy balanced efficient and reliable routing protocol for underwater wireless sensor networks, IEEE Access 7 (2019) 175980-175999.

[45] M. Ismail, M. Islam, I. Ahmad, F.A. Khan, A.B. Qazi, Z.H. Khan, Z. Wadud, M. Al-Rakhami, Reliable path selection and opportunistic routing protocol for underwater wireless sensor networks, IEEE Access (2020).

[46] Zhao. Ming, Arun. Kumar, Peter.Han.Joo. Chong, Rongxing. Lu, A reliable and energy-efficient opportunistic routing protocol for dense lossy networks, IEEE Wirel. Commun. Lett. 6 (1) (2016) 26–29.

[47] Z. Wadud, K. Ullah, I. Ahmad, A.B. Qazi, S. Jan, F.A. Khan, N. Minallah, An efficient routing protocol based on stretched holding time difference for underwater wireless sensor networks, Sensors 19 (24) (2019) 5557.