Wireless Pers Commun (2011) 57:607–627
DOI 10.1007/s11277-009-9864-3

# A Distributed Energy-Aware Routing Protocol for Underwater Wireless Sensor Networks

**Mari Carmen Domingo**

**Abstract** The design of routing protocols for Underwater Wireless Sensor Networks (UWSNs) poses many challenges due to the intrinsic properties of underwater environments. In this paper we present DUCS (Distributed Underwater Clustering Scheme), a new GPS-free routing protocol that does not use flooding techniques, minimizes the proactive routing message exchange and uses data aggregation to eliminate redundant information. Besides, DUCS assumes random node mobility and compensates the high propagation delays of the underwater medium using a continually adjusted timing advance combined with guard time values to minimize data loss. The theoretical and simulation studies carried out demonstrate its effectiveness.

**Keywords** Acoustic communications · Routing · Energy efficiency · Underwater networking

## 1 Introduction

The sea is a fascinating large expanse of water that has always attracted people who wanted to solve its mysteries. For centuries the access of human beings to the sea was limited to the surface or the nearby water, because the researchers had to use wire-line instruments and sampling equipment located at the sea surface. This fact restricted scientific research operations.

Nowadays there is a growing need of underwater monitoring (e.g. for exploration of natural undersea resources, gathering of scientific data or detection of marine incidents such as chemical pollution or oil spill) but the existing technologies do not measure up to the demanding requirements.

M. C. Domingo (✉)
Telematics Engineering Department, Barcelona Tech University, Esteve Terradas,
7, 08860 Castelldefels, Barcelona, Spain
e-mail: cdomingo@entel.upc.edu

Consequently, a new concept of low-cost, more easily deployable underwater networks with less restricted conditions should be developed: Underwater Wireless Sensor Networks (UWSNs) [1].

These kinds of networks should be scalable, mobile and capable of self-organization. They eliminate the need for cables and do not interfere with shipping activity [2].

UWSNs are a new research paradigm that poses exciting challenges compared to the ground-based existing networks due to the intrinsic properties of the underwater environments. They suffer from:

- Large propagation delays. The propagation speed of acoustic signals in water is about $1.5 \times 10^3$ m/s [3].
- Node mobility. Underwater sensor networks move with water current [4].
- High error probability of acoustic underwater channels. The underwater acoustic communication channel has a very limited bandwidth capacity (of the order of KHz), variable delays and suffers high bit error rates.

Therefore, the restricted network operation conditions pose a motivation for doing research at each layer of the protocol stack [1,5].

Energy saving is a major concern in UWSNs because sensor nodes are powered by batteries, which are difficult to replace or recharge in aquatic environments. The design of robust, scalable and energy-efficient routing protocols in this type of networks is a fundamental research issue. Most existing data forwarding protocols proposed for ground-based sensor networks [6,7] cannot be directly applied because they have been designed for stationary networks. The existing multi-hop ad hoc routing protocols are not adequate because they apply a continuous exchange of overhead messages (proactive ad hoc routing) or employ a route discovery process based on the flooding technique (reactive ad hoc routing); these mechanisms are inefficient tools in large scale underwater networking because they consume excessive energy and bandwidth resources.

In this paper we present DUCS (Distributed Underwater Clustering Scheme), a new distributed energy-aware routing protocol designed for long-term non-time-critical aquatic monitoring applications using UWSNs with random node mobility and without GPS (Global Positioning System) support. Our clustering protocol does not use flooding techniques, minimizes the proactive routing message exchange and it uses data aggregation to eliminate redundant information before transmission to the sink. Some examples of the long-term non-time critical aquatic monitoring applications where the implementation of our routing scheme could be very helpful are marine biology, deep-sea archaeology, seismic predictions or pollution monitoring.

We propose TDMA and CDMA (Code Division Multiple Access) with DSSS (Direct Sequence Spread Spectrum) using pseudo-orthogonal codes for intra-cluster communication and only CDMA with DSSS using pseudo-orthogonal codes for all other communications processes. Adjacent clusters to another cluster use different spreading codes, but scalability is achieved through spatial reuse of the same codes in non-adjacent clusters. In this way, intra-cluster interference is eliminated and inter-cluster interference is reduced. Besides, DUCS compensates the high propagation delays of the underwater medium using a continually adjusted timing advance combined with guard time values to minimize data loss and maintain communication quality.

We have validated our routing protocol through theoretical analysis and simulations.

The paper is structured as follows. Section 2 explains related work about the design of routing protocols in UWSNs. Section 3 describes in detail how our routing protocol works. Section 4 introduces the sound propagation model. Section 5 analyzes which is the optimum number of clusters for the routing protocol design. Section 6 shows our simulation results. Finally, Sect. 7 concludes this paper.

## 2 Related Work

Many authors have proposed hierarchical or cluster-based routing protocols [7–9] for ground-based sensor networks as a way to improve the scalability, lifetime and energy efficiency of the network. However, these protocols are not appropriate for UWSNs because they assume that the sensor network is stationary and they are not well adapted to the intrinsic properties of underwater environments, such as long propagation delays, low data rates and difficulty of synchronization. LEACH [7] was the first hierarchical routing approach for ground-based sensor networks. In LEACH a set of sensor nodes are selected as cluster-heads and the cluster-head role is rotated to spread the energy dissipation to all nodes in the network. Besides, data-aggregation techniques are employed, and TDMA/CDMA is used to reduce interference and collisions. However, this solution requires all cluster-heads to reach the sink to send the aggregated data, an assumption that strictly limits the network area and applications. Therefore, it cannot be useful in UWSNs because these networks can be deployed in large regions. Besides, the cluster-head selection algorithm assumes that all nodes have the same energy resources (maximum battery capacity). In addition, this routing approach is not well-suited for mobile networks. On the contrary, DUCS does not suffer such limitations because multi-hop routing between clusters and an energy-aware cluster-head selection algorithm based on information sent by neighbor nodes are used. Furthermore, it incorporates a cluster maintenance phase to extend the protocol usage to mobile networks such as UWSNs and defines timing advance and guard time values to adapt properly to the intrinsic characteristics of underwater environments and to improve data reception.

Finally, some routing protocols [3,10–14] have been specifically designed for UWSNs. Some of them are location-based [3,10–12]; In [3,10] the authors use the concept of routing vector (defined as a vector from the source to the sink [3] or as a vector for each single forwarder (hop-by-hop vectors) [10]); In [11] the authors take into account the varying conditions of the underwater channel and the type of sensor network applications and design algorithms for delay-sensitive or delay-insensitive routing. In [12] a scalable routing technique is proposed that optimizes the minimum energy per bit consumption. Although location information is needed for these geo-routing protocols, localization is still a challenging research issue [15,16].

Another routing protocol [13] tries to increase the probability of successful delivery forwarding data over more routes towards different local sinks which collectively form a virtual sink (multipath routing); the packet delivery ratio is improved but the energy increase has not been analyzed. In [14] a routing protocol has been proposed with no proactive routing message exchange and negligible amount of on-demand floods; although this protocol minimizes the number of on-demand floods, no energy consumption study is provided; in addition, the simulations carried out use pure ALOHA as contention avoidance method, a low efficient protocol whose performance is highly affected by the propagation delay, and the retransmission of lost packets increases the power consumption and diminishes the network survivability. On the contrary, CDMA has been advocated for underwater environments due to its properties (efficient use of bandwidth, multipath resolution, robustness to interference [1,5,17]) and we think that the research efforts should concentrate on techniques to improve the possible limitations of this protocol (e.g. spatial reuse of the spreading codes or optimization of the transmit power and code length [18]).

All these different protocols have some common characteristics: They assume GPS-free nodes; besides, they try to be adaptive, scalable and energy-efficient, some fundamental properties for the design of routing protocols in this type of networks. However, the theoretical analysis carried out in [19] demonstrates that the routing protocols based on the clustering scheme save more energy and they show a better performance in shallow water (with sea depth lower than 100 m). In addition, it has been demonstrated that the clustering scheme is scalable with respect to the number of sensor nodes and the distance between them. Therefore we focus on the design of cluster-based routing protocols for UWSNs.

In [20] the authors propose another clustering protocol that uses TDMA/CDMA for network communication. However, this solution is not applicable to UWSNs because it assumes that cluster formation and maintenance is based on the nodes' position and movement information received using cables and GPS. DUCS, on the other hand, does not suffer such important limitations because it is well-suited for wireless sensor nodes and able to operate in a GPS-free network. Furthermore, DUCS incorporates an energy-aware cluster-head selection algorithm, as well as data aggregation, to eliminate redundant information (we assume an application such as pollution monitoring, where the nodes inside a cluster are close enough to send the same or very similar data at certain times), and a timing advance technique to compensate for the large underwater propagation delays.

We presented a preliminary version of the proposed distributed clustering scheme for the shallow water scenario [21]. In the present paper an analytical study has been also carried out. In addition, we discuss in more detail the decisions on the design of this approach and the pros and cons of the proposed solution. In [22] the good performance of our routing protocol has been demonstrated for the deep water scenario (with sea depth larger than 100 m).

## 3 DUCS Protocol

### 3.1 Protocol Architecture

DUCS is an adaptive self-organizing protocol where clusters are formed using a distributed algorithm. We consider an application where underwater sensor nodes have always data to be sent to the sink (e.g. pollution monitoring) and the sensor nodes can use power control to adjust their transmission power. What is more, they have virtual IDs of small size (a couple of bytes). These IDs should be unique in the sensor network (not globally but locally unique).

The operation of DUCS is illustrated in Fig. 1. The nodes organize themselves into local clusters, and one node is selected as a single cluster-head for each cluster. All non-cluster head nodes transmit their data to their cluster-head via a single hop; the cluster-head node receives data from all cluster members, performs signal processing functions on the data (e.g. data aggregation) and transmits it to the sink using multi-hop routing (relaying it through other cluster-heads, each cluster-head can do power control). Frequently, nodes close to each other process highly correlated data because they monitor the same phenomena, and with the aid of data aggregation techniques the effective non-redundant data can be extracted by the cluster-head and sent to the sink, thus saving energy and bandwidth. Cluster-heads are responsible for coordination among nodes within their clusters (intra-cluster coordination) and for communication with other cluster-heads (inter-cluster communication).

DUCS incorporates randomized rotation of the cluster-head among the sensors in order to avoid fast draining the batteries of specific underwater sensors in the network. In this way, energy consumption is more evenly distributed among the different nodes.

The operation of DUCS is divided into rounds (see Fig. 2).

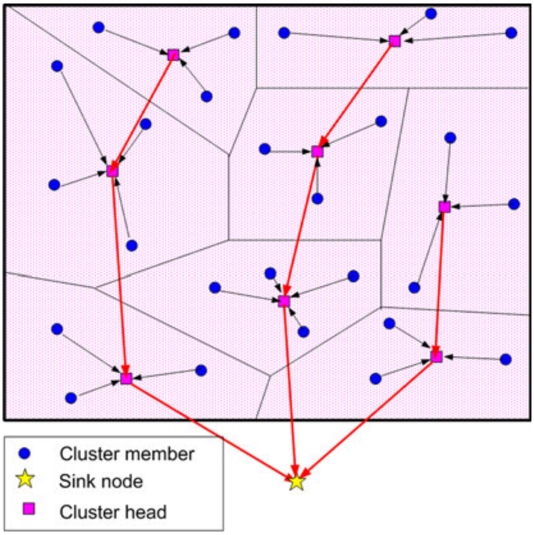


**Fig. 1** Clustering using DUCS

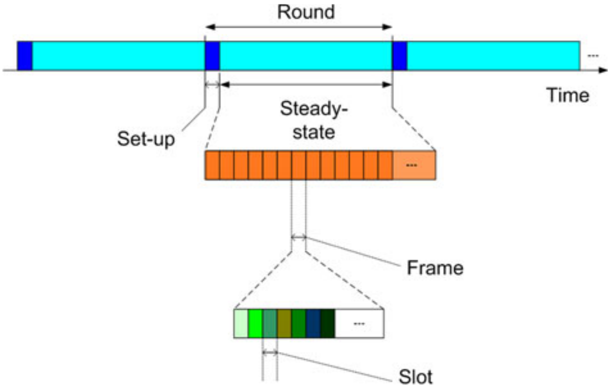


**Fig. 2** Time line of DUCS

All clusters are formed during the set-up phase or cluster creation process and data transfer occurs during the steady-state or network operation phase.

During the network operation phase several frames are sent to each cluster-head; a frame is formed by a series of data messages that non-cluster head sensor nodes send to the cluster-head using a schedule (each non-cluster head sensor node sends one data message consuming a time slot). Afterwards, both phases are repeated periodically. One must ensure that the network operation phase is long compared to the cluster creation process in order to minimize the overhead and improve the performance of the routing protocol.

## 3.2 Cluster-Head Selection Algorithm

The cluster-head selection algorithm is used to elect a certain number of cluster-heads during the cluster creation process.

A node initially sets its probability to become cluster-head as follows:

$$CH_{prob} = C_{prob} \tag{1}$$

where $C_{prob}$ is a small constant fraction used to set an initial percentage of cluster-heads; thus the number of announcements from nodes electing themselves as cluster-heads is reduced. If a node elects itself as cluster-head, it sends an advertisement message containing its virtual ID to its neighbours using CDMA.

Afterwards, the value of $CH_{prob}$ is computed as follows. During the cluster creation process, the nodes compute their remaining energy and calculate their probability of becoming cluster-heads, $CH_{prob}$. If $CH_{prob}$ falls above a random value between 0 and 1, a node elects itself as cluster-head. During this phase, a node processes the cluster-head announcements it has received to select the lowest cost cluster-head. The communication cost can be defined as a function of the distance between source and destination. Each non-cluster head decides to which cluster it belongs by choosing the cluster-head that requires the minimum power level for transmission (we assume that a node can do power control) and, consequently, minimizes the communication energy. The transmission power is directly proportional to the distance between sensor nodes in shallow water scenarios. Therefore, each non-cluster head should calculate its distance (cost) to each self-elected cluster-head neighbour with the aid of acoustic-only time-of-arrival (ToA) approaches (e.g. measuring round-trip time that an acoustic signal suffers) [1] and select the nearest one. For this purpose a ping request is sent as a unicast message containing the virtual ID of the sender (non-cluster head). A ping reply is sent back as a unicast packet towards the non-cluster head. After each node has decided to which cluster it wants to belong, it must inform the cluster-head. Each node transmits a join-request message back to the chosen cluster-head using CDMA. This message is short and consists of the node's ID and the cluster-head's ID. A node that doesn't receive any cluster-head announcement and does not have the sink as neighbour is not allowed to send data messages during this round. However, this situation is very uncommon, since the expected number of cluster heads is designed taking into account the network density.

After the first round is over, a node sets its probability of becoming cluster-head during the cluster creation process as follows:

$$CH_{prob} = \min\left\{\frac{C_i}{E_{AV}} \times C_{prob}, 1\right\} \tag{2}$$

where $C_i$ represents the node's battery level (residual energy of a sensor node) and $E_{AV}$ means the average energy per cluster and is calculated as $\frac{\sum_{i=1}^{S_j} C_i}{S_j}$ for $S_j$ cluster members in a cluster $j$.

After each round before the cluster creation process each cluster member should send to the cluster-head its residual energy value $C_i$ using TDMA. The cluster-head uses this information to calculate $E_{AV}$ and sends this parameter value as a broadcast message to its cluster members; thus, each cluster member is able to calculate its own $CH_{prob}$ for the next round. In this way we ensure that the number of cluster-heads does not diminish when the network energy resources are scarce.

Since each node becomes a cluster-head with probability $CH_{prob}$ and the number of nodes in the network is $N$, the expected number of cluster-head nodes per round using this algorithm and assuming that the battery energy of a node is very similar to the average battery energy of a cluster is:

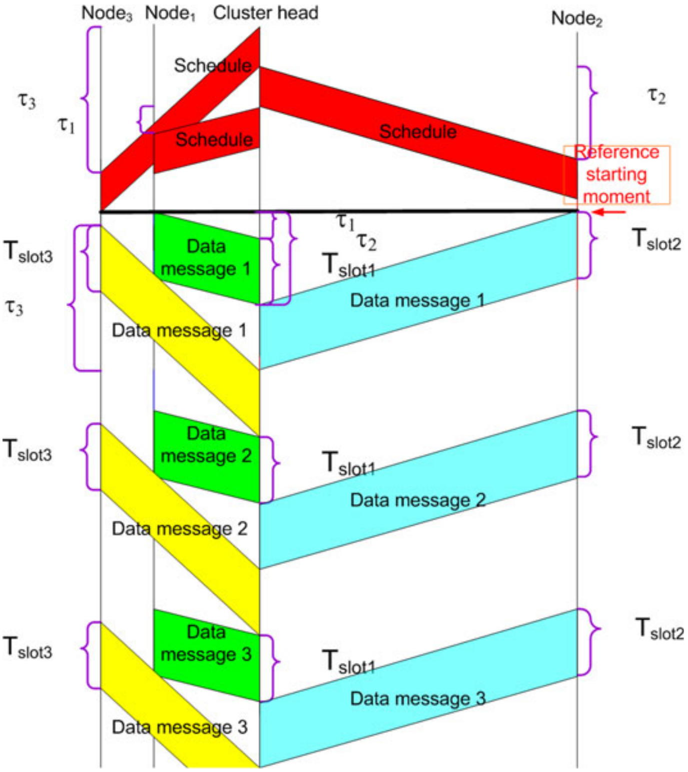


**Fig. 3** Example of a time diagram in a cluster with three members (in addition to the cluster-head)

$$z = E\left[N^{o}CH\right] = \sum_{i=1}^{N} CH_{prob_i}(t) = N \times C_{prob} \tag{3}$$

where $CH_{prob_i}(t)$ is the probability that a node elects itself as cluster-head.

### 3.3 Cluster Formation Algorithm and Network Operation Phase

In Fig. 3 we can find an example of the cluster formation algorithm and network operation phase in a cluster with a cluster-head and three non-cluster head nodes.

After the cluster creation process is over, each cluster-head knows which nodes belong to its cluster. Now the cluster-head should coordinate the data transmissions in its own cluster. The cluster-head sets up a TDMA (Time-Division Multiple Access) schedule and transmits this schedule using CDMA to the cluster members (see Fig. 3). TDMA has been selected as medium access control (MAC) protocol inside a cluster because it avoids collisions between non-cluster head members of the same cluster and because it enables that non-cluster head nodes are turned off whereas they do not transmit; therefore, they remain in the sleep mode and thus energy consumption is reduced.

An important problem in underwater communications is the fact that data messages from different cluster members could overlap at the cluster-head because of their different high propagation delays in the underwater medium, resulting in poor transmission quality or even in communication loss. Our proposed solution is that each sensor node advances its transmission relatively to its reception by a time compensating the propagation delay. This value is called timing advance, a concept used in other communications systems like GSM (Global System for Mobile Communications) [23]. The timing advance value for each node can be computed only by the cluster-head, and is then forwarded to the underwater sensor nodes included in the TDMA schedule.

When a cluster-head knows which nodes will belong to its cluster, it sends an acoustic signal to them in order to measure the round-trip time and as a result to estimate the propagation delay to each non-cluster head node in its cluster with the aid of ToA techniques [1].

Suppose that nodes $N_1, N_2, N_3, \ldots, N_f$ have joined the cluster with cluster-head node $CH_j$; then the cluster-head sends acoustic signals to know the propagation delays from itself to each cluster member, which are $\tau_1, \tau_2, \tau_3, \ldots, \tau_f$, respectively.

The nodes are classified according to their propagation delays from the largest to the lowest ones and the schedules should be sent in this order to decrease the delay. The cluster-head knows that once the schedule has been sent to the cluster members, the node $N_l$ with the largest propagation delay will receive the schedule only after $\tau_l$; therefore it establishes (once the schedules have been sent) a starting moment for transmission after $\tau_l$ (reference starting moment); this means that nodes $N_1, N_2, N_3, \ldots, N_f$, which receive their schedules only after $\tau_1, \tau_2, \tau_3, \ldots, \tau_f$, should wait once they have received their schedules for $\tau_l + t_{schedule} - \tau_1 - x_1 \cdot t_{schedule}, \tau_l + t_{schedule} - \tau_2 - x_2 \cdot t_{schedule}, \tau_l + t_{schedule} - \tau_3 - x_3 \cdot t_{schedule}, \ldots, \tau_l + t_{schedule} - \tau_f - x_f \cdot t_{schedule}$, where $t_{schedule}$ is the transmission time of the schedule and $x_i$ is a variable that indicates the number of $t_{schedule}$ a node has to wait according to the transmission order; thus, we ensure that all nodes adjust the same reference starting moment at the same time.

Frames are sent to the cluster-head; frames are divided into time slots, and each time slot of the same frame is occupied to transmit a data message of a different node; the duration of a frame is fixed to $k \times T_{slot}$, where $k$ represents the number of time slots of a frame and this value is set to the number of non-cluster head members (the number of cluster members minus the cluster-head); the value $k$ varies depending on the number of non-cluster head members of each cluster; if there are on average $\frac{N}{z} = \frac{N}{N \times C_{prob}} = \frac{1}{C_{prob}}$ nodes per cluster, $k = \left(\frac{1}{C_{prob}} - 1\right)$.

A round can be defined as a period of time where clusters are organized and frames are transmitted from the cluster members to the self-elected cluster-head in each cluster. A frame is formed by a series of data messages that the non-cluster head sensor nodes send to the cluster-head using a schedule (each non-cluster head sensor node sends one data message consuming a time slot). When the start of the first frame begins, the number of turns $t$ is initialized to 0 and each node should send its data message in this first turn. The nodes are classified according to their propagation delays from the lowest to the largest ones and they should send information in this order, because thereby the cluster-head does only need to wait for $\tau_s$ s (lowest propagation delay) the transmission of the first data message and from here on data messages are sent uninterruptedly. This means that in the schedule the node with the minimum propagation delay $\tau_s$ should start sending a data message at $T_{start_s}$ (reference starting moment) consuming a time slot. The node with the second minimum propagation delay is the next and so on. With this transmission order some extra delay is saved.

The second node $N_{s+1}$ should start the transmission of its first data message at:

$$T_{start_{s+1}} = \tau_s + T_{slot_s} - \tau_{s+1} \tag{4}$$

where $T_{slot_s}$ stays for the transmission time (using one time slot) of the first data message from node $N_s$.

The third node $N_{s+2}$ should start its transmission at:

$$T_{start_{s+2}} = \tau_s + T_{slot_s} + T_{slot_{s+1}} - \tau_{s+2} \tag{5}$$

In general, the node $N_{s+i}$ should start the transmission of its first data message at:

$$T_{start_{s+i}} = \tau_s + \sum_{j=s}^{s+i-1} T_{slot_j} - \tau_{s+i} \tag{6}$$

Finally, the last node $N_{s+u}$ transmits its first data message. When all nodes have sent one data message, a complete frame has been transmitted; the number of turns $t$ is increased to 1, $i$ is initialized to 0 for each new turn and the process is repeated in the same transmission order. In general, the node $N_{s+i}$ should start the transmission of its $(t+1)$-th data message in a turn at:

$$T_{start_{s+i}} = \tau_s + t \times \sum_{j=s}^{s+u} T_{slot_j} + \sum_{j=s}^{s+i-1} T_{slot_j} - \tau_{s+i} \tag{7}$$

where $u$ represents the number of nodes transmitting minus 1.

We assume that the transmission times of the data messages $T_{slot_j}$ are large enough so that every starting transmission moment $T_{start_{s+i}} \geq 0$.

So far we have assumed that the propagation delay remains the same for each data message sent by a particular sensor node, but the propagation delay in reality varies due to channel fluctuations caused by the relative motion of the transmitter, receiver, or significant scattering surfaces [24]. Therefore, in order to avoid acoustic collisions at the cluster-head when two non-cluster head members using adjacent time slots send their data messages, we have decided that each node places a period called "guard time" of its transmission duration. This period allows the information to reach a certain distance without any interference caused by the subsequent transmission. The establishment of a timing advance reduces considerably the time guard length, which can be calculated as follows. The members of a cluster should be synchronized. Once the schedules have been sent by the cluster-head, two main issues concerning time synchronization should be considered:

1. Use of independent clocks by each sensor node.
2. Node mobility varies the distance between sensor nodes.

The first issue will not be significant if the guard time values are properly designed to absorb quartz clock drifts. Because new schedules are generated at each round, the clock drift does not accumulate over rounds. The quartz clock drift is 1 s every 1.000.000 s. This means that the guard time value should be larger than $\frac{T_{round}}{1.000.000}$, where $T_{round}$ means time between rounds. In our simulations we set $T_{round} = 200\,\mathrm{s}$ and $T_{slot} = 80\,\mathrm{ms}$. Therefore the clock drift would be 0.2 ms and thus the time guard length is set to $\frac{T_{slot}}{100} = 0.8\,\mathrm{ms}$.

The second issue related to synchronization is addressed in Sect. 3.5.

After the transmission of a frame to the cluster-head, one time slot is used for cluster maintenance; due to node mobility the nodes positions vary with time and it is necessary to modify the cluster members and TDMA schedules accordingly. The cluster maintenance algorithm is explained in Sect. 3.5.

### 3.4 Multi-Hop Routing Between Cluster-Heads

Once the cluster-head has received a data message from each cluster member, it performs signal processing functions to compress this data together with its own data message into a single signal. The composite signal is sent to the sink though other cluster-heads using CDMA and multi-hop routing. We have selected as metric for multi-hop routing the distance between cluster-heads because the underwater sensor nodes can do power control and adjust their power levels related to the receiver's distance. Thus energy is saved. The multi-hop routes are created as follows:

After the cluster creation process is over, the sink calculates its distance to its cluster-head neighbours using ToA techniques (the cluster-head sends an acoustic signal as a unicast ping request containing its virtual ID and the cluster-head neighbour nodes send a reply as a unicast packet). Finally, the sink sends the distance cost to its cluster-head neighbours as a unicast message. This means that each cluster-head neighbour $CH_{j \in V}$ will know distance $CH_i - S$, where $V$ is the set containing all cluster head nodes that are neighbours from the sink $S$. After having received this message, each cluster-head neighbour from the sink ascertains its distance to its cluster-head neighbours with the aid of ToA techniques and computes distance $CH_i - S =$ distance $CH_i - CH_j +$ distance $CH_j - S$, where $CH_{i \in L}$ and $L$ is the set containing all cluster-head nodes neighbours from $CH_{j \in V}$. We consider that distance $CH_i - S$ means the distance from node $CH_i$ towards the sink through the intermediate cluster-head $CH_j$ and not directly. Each cluster-head $CH_i$ that receives routing packets with the distance cost from different cluster-head neighbours as a unicast message should select as next hop to forward its frames towards the sink $S$ the adjacent cluster-head $CH_j$ that minimizes:

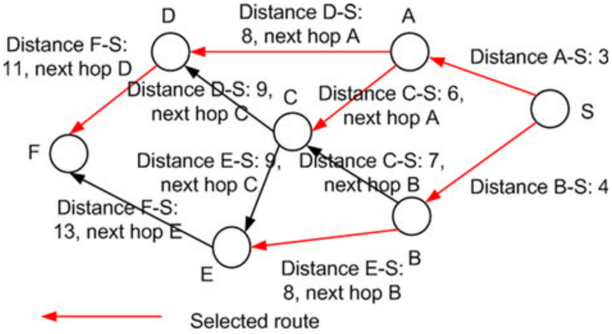
$$\operatorname{argmin}\left(\text{distance } CH_i - CH_j + \text{distance } CH_j - S\right).$$

The node $CH_i$ should calculate its distance to its neighbours with the aid of ToA techniques, too and send them the distance cost from its neighbours towards the sink using $CH_i$ as relay. The process should be repeated by each cluster-head in the same way. The routing algorithm for selecting multi-hop routes between cluster-heads should be repeated after the cluster-creation process is over once in each round in order to avoid excessive signalling. However, each cluster-head should periodically use ToA techniques to check if the cluster-head neighbours are still reachable or, on the other hand, if its distance towards them is $\gamma\%$ higher than the last measure. In this case this cluster-head should warn the sink and the sink should initiate the route discovery phase. Only the sink can start the route discovery process. If the sink has initiated a new route discovery process and a cluster-head receives new distance costs, it should again recalculate its best path towards the sink and forward the new distance costs towards its neighbours. It should be mentioned that a cluster-head can simultaneously receive packets from different cluster-head neighbours because every cluster-head adjacent to another cluster-head uses a different spreading code and each cluster-head has a bank of matched-filter correlators to obtain the data from different spreading codes (the sink uses its own spreading code, too). Figure 4 shows a multi-hop routing example between cluster-heads and a sink $S$. In our simulations $\gamma$ is set to 50%.

### 3.5 Cluster Maintenance

The nodes in underwater acoustic environments move and consequently the cluster organization is affected as well as the TDMA schedules. The cluster maintenance algorithm works as follows:

During the time interval reserved for maintenance purposes each node should again estimate its propagation delay towards its cluster-head with the aid of ToA techniques. For this purpose each cluster-head should send a broadcast packet periodically to maintain the required synchronization of TDMA. As we see in Fig. 5a the cluster-head broadcasts a SYNC packet to all its cluster members. The SYNC packet announces the cluster-head period between



| Selected routes: | Distance cost: |
|---|---|
| A-S | 3 |
| B-S | 4 |
| C-A-S | 6 |
| D-A-S | 8 |
| E-B-S | 8 |
| F-D-A-S | 11 |

**Fig. 4** Multi-hop routing between cluster-heads

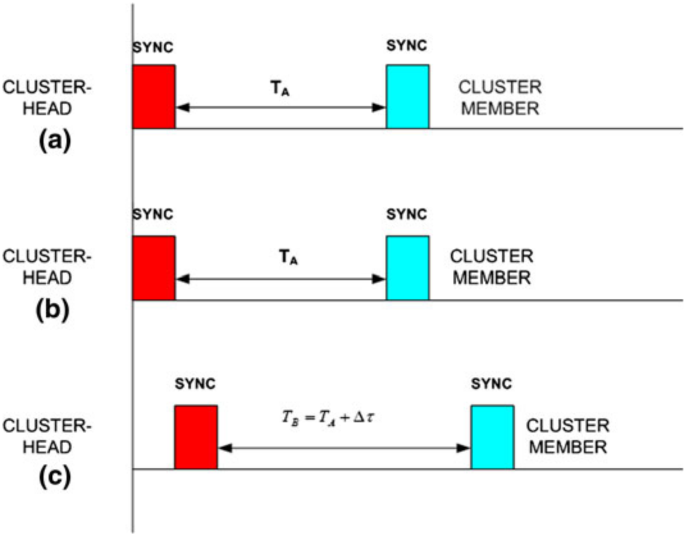


**Fig. 5** Cluster maintenance phase

SYNCs $T_A$. A cluster member listens to the channel to achieve frame synchronization with the cluster-head. The reception of the SYNC packet by the cluster member with the explicit stamping of the period between SYNCs, $T_A$, enables it to adjust the time to wake up in the next cycle to listen to the cluster-head; this procedure works fine as long as the propagation delay remains fixed (see Fig. 5b); however, if the distance between the cluster-head and the non-cluster head node varies due to mobility, the non-cluster head node will receive the SYNC after $T_B$, where $T_B = T_A \pm \Delta\tau$ and $\Delta\tau$ refers to the propagation delay variation (see Fig. 5c).

For this reason, after having received the SYNC, each non-cluster head member should recalculate again its delay to the cluster-head as $T_B = T_A \pm \Delta\tau$. If $T_B$ is not $\beta\%$ higher than the delay $T_A$ previously computed, the non-cluster head member should again recalculate the transmission of its data messages $T'_{start_{s+i}} = T_{start_{s+i}} \pm \Delta\tau$. If the distance between the cluster-head and a non-cluster head node varies due to mobility, the non-cluster head should be awake (idle listening mode) to listen to the SYNC in the worst case during $\Delta T_c$, which can be calculated as $2\frac{N}{z}T_{slot}\frac{v_{mobility}}{v_{prop}} = 2\frac{1}{C_{prob}}T_{slot}\frac{v_{mobility}}{v_{prop}}$, where $v_{prop}$ means propagation speed and $v_{mobility}$ is the node mobility speed. This means that the cluster maintenance phase should last at least $\Delta T_c$. We consider that a cluster maintenance phase duration of one time slot will be enough. If a node listening hears a signal, it changes its state to receive mode.

If the delay $T_B$ is $\beta\%$ higher than the same parameter calculated during the set-up phase, the node should again estimate the distance to each cluster-head neighbour that had announced itself during the set-up phase using ToA techniques and with this information it should select the closest cluster-head neighbour. If this cluster-head is a different one as the previously selected cluster-head, it should send a message to join the new cluster and another message to the previous cluster-head to leave the cluster. Then the affected cluster-heads should again recalculate the TDMA schedules and send them to their cluster members. $\beta$ is set to 50%.

### 3.6 Intra-Cluster and Inter-Cluster Communication

We propose TDMA and CDMA (Code Division Multiple Access) with DSSS (Direct Sequence Spread Spectrum) using pseudo-orthogonal codes [25] for intra-cluster communication (communication inside a cluster) and only CDMA with DSSS using pseudo-orthogonal codes in all other communications processes.

Each node that elects itself as cluster-head, selects randomly a unique spreading code to send an advertisement message to its neighbours. If a cluster-head candidate $i$ has just listened to a previous advertisement message from another cluster-head with a specific spreading code $\delta_j$, it should use another different unique spreading code when announcing itself as cluster-head $\delta_i$. If a node receives two advertisement messages from two different cluster-heads using the same spreading code $\delta = \delta_i = \delta_j$, the node should advertise one of them about the situation so that this cluster-head $i$ should resend the advertisement message with a new different spreading code $\delta_i$ that invalidates the previous one. If there are no more different spreading codes available, it is not possible that this node declares itself as cluster-head. One of the advantages of using a clustering protocol is that we need a different spreading code per cluster and not per node; consequently, the number of codes is not so fast exhausted. Adjacent clusters to another cluster use different spreading codes and scalability is achieved by spatial reuse of the same codes.

When a node replies to the advertisement message of a particular cluster-head to join this cluster, it uses the same spreading code as the cluster-head previously did. All non-cluster heads transmit their data to the cluster-head using TDMA and the same spreading code, and again the same code is used by the cluster-head when it sends the aggregated data to the first hop towards the sink using multi-hop routing.

Figures 6 and 7 illustrate an example of code assignment among clusters. Figure 7 shows the frame structure for the network partitioning shown in Fig. 6.

The advantages of using CDMA/TDMA are that intra-cluster interference is eliminated and inter-cluster interference is reduced with a transmitter-based code assignment. Intra-cluster interference is eliminated because non-cluster head nodes transmit in order using a TDMA schedule. Inter-cluster interference is reduced because adjacent clusters to another cluster use a different spreading code for transmission.

**Fig. 6** Example network with DUCS

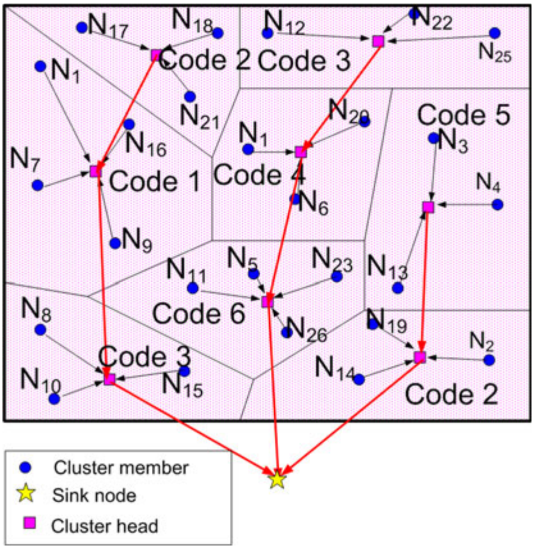
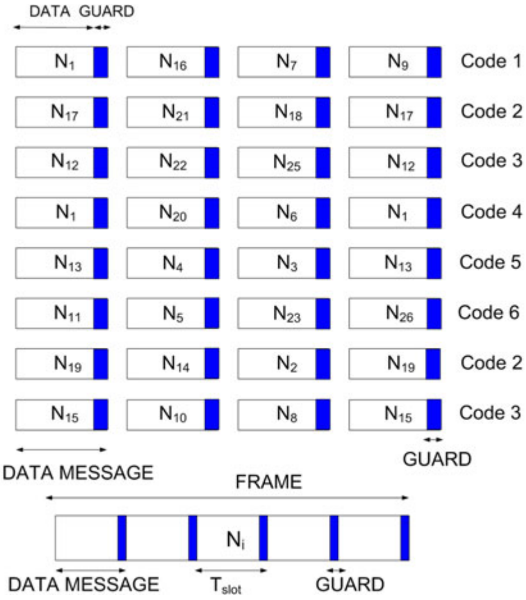


**Fig. 7** Frame structure and code assignment



## 3.7 Data Aggregation

We introduce a complementary exponential data correlation model based on [26]. We assume that each node inside a cluster collects $l$ bits and sends them back to its cluster-head at distance $r$; the cluster-head receives one data message from each node and spends $\frac{N}{z} l E_{DA}$ Joules to perform data aggregation on the $\frac{N}{z} l$ bits (collected by itself and their members), where $\frac{N}{z} = \frac{1}{C_{prob}}$ represents on average the number of nodes per cluster and $E_{DA}$ means the energy spent in data aggregation per bit and is set to 5 nJ/bit. The resulting data has a size of $l\left(1+\left(\frac{N}{z}-1\right)\eta\right)$, where $\eta$ is the data aggregation residual ratio and is assumed to be complementary exponential, namely,

$$\eta = 1 - e^{-\alpha r} \tag{8}$$

where $r$ represents the distance between two sensor nodes and $\alpha$ is a positive number whose value depends on a specific event of interest. It is accomplished that $0 < \alpha < 1$. In our research study $\alpha$ is set to 0.005 and the value of $r$ is equal to the distance between the cluster-head and the remotest node in this cluster (reduced data aggregation).

## 4 Sound Propagation Model

When an underwater sensor transmits a packet of $l$ bits at distance $d$, it spends:

$$E_{TX}(l, d) = l E_{ELEC} + E_{TX-SOUND}(l, d) \tag{9}$$

where $E_{ELEC}$ refers to the energy per bit needed by transmitter electronics and digital processing and $E_{TX-SOUND}$ refers to the energy spent in the transmission of underwater acoustic signals.

We concentrate on the shallow water scenario (water with depth lower than 100 m). In this scenario $E_{TX-SOUND}$ can be expressed as [27]:

$$E_{TX-SOUND}(l, d) = P T_{TX} = 2\pi d H I_1 T_{TX} \tag{10}$$

where $P$ represents the power level consumed by a node during transmission, $I_1$ is the intensity at a distant point in the sea, $T_{TX}$ represents the transmission time for one packet, $d$ means the transmission range and $H$ is the sea depth. $T_{TX} = \frac{l}{v_{TX}}$, where $v_{TX}$ means transmission speed.

The signal to noise ratio (SNR) of an emitted underwater signal at the receiver can be expressed by the passive sonar equation [27]:

$$SNR = SL - TL - NL + DI \geq DT \tag{11}$$

where $DT$ has been defined as the detection threshold, $SL$ is the target source level or noise generated by the target, $TL$ is the transmission loss due to the water environment, $NL$ is the noise level (from the receiver + the environment) and $DI$ is the directivity index (a function of the receiver's directional sensitivity).

The sonar parameter $TL$ can be defined as the accumulated decrease in acoustic intensity as an acoustic pressure wave propagates outwards from a source. This magnitude can be estimated by adding the effects of geometrical spreading, absorption and scattering [27].

If $I_0$ is the intensity at the reference point located 1 *yard* (1 *yd* = 0.9144 m) from the "acoustic center" of the source and $I_1$ is the intensity at a distant point in the sea, then [27]:

$$TL = 10 \log \frac{I_0}{I_1} = 10 \log I_0 - 10 \log I_1 \tag{12}$$

For frequencies below 500 KHz transmission loss is mainly caused by *spreading effects* and the impact of absorption and scattering is negligible [14]:

$$TL = 10 \log d \tag{13}$$

The source level $SL$ can be defined as the intensity of the radiated sound in decibels related to the intensity of a plane wave of root mean square (rms) pressure 1 $\mu$Pa, referred to a point *1 yd* (0.9144 m) from the "acoustic center" of the source in the direction of the target [27]:

$$SL = 10 \log \frac{I_0}{I_{ref}} = 10 \log \frac{I_0}{1\,\mu\text{Pa}} \tag{14}$$

## 5 Optimum Number of Clusters

In DUCS we consider that the expected number of clusters per round is $z$. Now we want to analytically determine the optimum value of $z$ that will minimize the energy dissipation in the system.

Suppose there are $N$ nodes distributed uniformly in a $S \times S \times H$ space, where $H$ represents the sea depth. If there are $z$ clusters, there are on average $\frac{N}{z}$ nodes per cluster.

Each cluster-head dissipates energy receiving signals from the nodes, doing data aggregation, and transmitting the aggregate signal to the sink through other cluster heads. We consider the worst case in terms of energy cost for the cluster-head, when the cluster-head can not use any other cluster-head as relay and it should send its frames directly to the sink. For this worst case the optimum value of clusters will be the same using DUCS or LEACH. Each node transmits a message of $l$ bits. Therefore, the energy dissipated in the cluster-head node during the transmission of a single frame, after each cluster member has transmitted a single data message is:

$$E_{CH} = lE_{elec}\left(\frac{N}{z} - 1\right) + lE_{DA}\left(\frac{N}{z}\right) + 2\pi d_{CH\ to\ SINK} H I_1 \frac{l\left(1 + \left(\frac{N}{z} - 1\right)\eta\right)}{v_{TX}} \tag{15}$$

where $d_{CH\ to\ SINK}$ is the distance from the cluster-head to the sink.

Each non-cluster node does only need to transmit its data to the cluster-head once during a frame transmission and transmits a single data message. Thus, the energy consumed in each non-cluster head node is:

$$E_{non-CH} = lE_{elec} + 2\pi d_{non-CH\ to\ CH} H I_1 \frac{l}{v_{TX}} \tag{16}$$

where $d_{non-CH\ to\ CH}$ is the distance from the node to the cluster-head.

We approximate for calculation purpose the volume occupied by each cluster to $\frac{S^2 H}{z}$. This is a volume with a node distribution $f(x, y, z)$. The expected distance from the nodes to the cluster-head (assumed to be at the center mass of the cluster) is given by:

$$\begin{aligned} E\left[d_{non-CH\ to\ CH}\right] &= \int\int\int \left(\sqrt{x^2 + y^2 + z^2}\right) f(x, y, z)\, \mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &= \int\int\int r f(r, \theta, \phi)\, r^2 sen\phi\, \mathrm{d}r\mathrm{d}\theta\mathrm{d}\phi \end{aligned} \tag{17}$$

We assume that this region is a cylinder with radius $R = \frac{S}{\sqrt{\pi z}}$ and $f(r, \theta, \phi)$ is constant for $r, \theta$ and $\phi$.

Equation (17) simplifies to:

$$E\left[d_{non-CH\ to\ CH}\right] = f \int_{\phi=0}^{\pi} \int_{\theta=0}^{2\pi} \int_{0}^{\frac{S}{\sqrt{\pi z}}} r^3 sen\phi\, \mathrm{d}r\mathrm{d}\theta\mathrm{d}\phi = f \frac{S^4}{\pi z^2} \tag{18}$$

If the density of nodes is uniform throughout the cluster area, then $f = \frac{1}{\frac{S^2 H}{z}}$ and

$$E\left[d_{non-CH\ to\ CH}\right] = \frac{S^2}{\pi \mathrm{Hz}} \tag{19}$$

Therefore,

$$E_{non-CH} = lE_{elec} + 2\frac{S^2}{z} I_1 \frac{l}{v_{TX}} \tag{20}$$

The energy dissipated in a cluster for the transmission of a frame is:

$$E_{cluster} = E_{CH} + \left(\frac{N}{z} - 1\right) E_{non-CH\ to\ CH} \tag{21}$$

The total energy during a frame transmission in the whole network if $E_{elec}$ and $E_{DA}$ are considered negligible is:

$$E_{total} = zE_{cluster} = 2\pi d_{CH\ to\ SINK} zHI_1 \frac{l\left(1 + \left(\frac{N}{z} - 1\right)\eta\right)}{v_{TX}} + (N - z)\, 2\frac{S^2}{z} I_1 \frac{l}{v_{TX}} \tag{22}$$

We can find the optimum number of clusters by setting the derivative of $E_{total}$ with respect to $z$ to 0. Therefore:

$$z = \pm S\sqrt{\frac{N}{\pi d_{CH\ to\ SINK} H(1 - \eta)}} \tag{23}$$

Figure 8 shows the total energy spent during frame transmission in the whole network according to Eq. (22). We can observe that the total energy spent during frame transmission increases as the distance from the cluster-head to the sink grows and it is increased for a very small number of clusters or when the number of clusters grows as well.

## 6 Simulations

We have run simulations with the NS-2 [28] tool to investigate the performance of our proposed approach. Therefore, we have modified the physical and MAC layers to support underwater communications. We have modelled the sound propagation model as described in Sect. 4.

The chosen scenario consists of $N$ mobile sensor nodes that communicate with a sink. The underwater sensor nodes are uniformly distributed in a volume of $S \times S \times H\,\mathrm{m}^3$. In order to study the scalability of the UWSN, the number of nodes $N$ varies from 50 to 250; when $N = 100$ the volume is $75 \times 75 \times 75\,\mathrm{m}^3$ and this volume is increased or diminished when $N$ varies to maintain the same node density. We use a random walk mobility model with a speed of 1.5 m/s. The sink is located at (50, 50, 0). Data messages are sent with a rate of $v_{TX} = 7$ Kbit/s, which is the payload data rate of the UWM1000 LinkQuest Underwater Acoustic Modem [29]. We have run 30 simulations for 1,000 s to assess and compare the performance of LEACH [7] with our proposed routing protocol in terms of packet delivery ratio (percentage of data packets successfully delivered), average routing overhead, average throughput and number of nodes alive per amount of data messages sent that arrive to the sink.

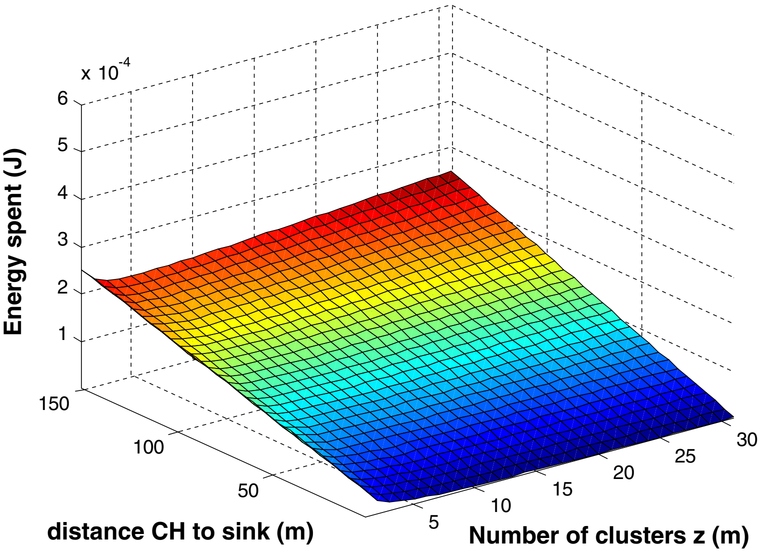


**Fig. 8** Total energy spent during frame transmission in the underwater network

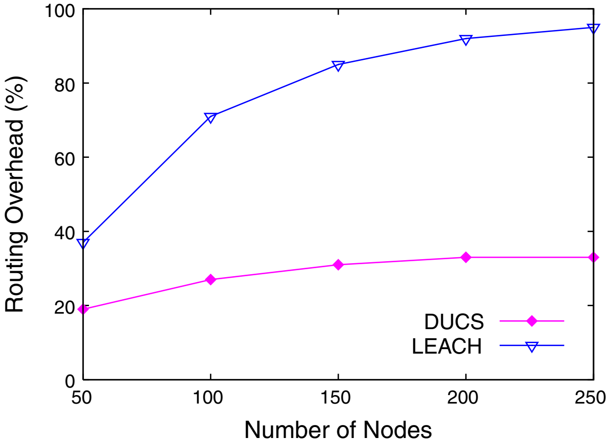


**Fig. 9** Routing overhead as a function of network size

In LEACH the number of clusters per round is set to N/5. The duration of a round for both routing protocols is set to 200 s. In our simulations the DSSS system has a 5 Megachips per second code clock rate, thus $G = 714.3$. An outage event occurs when, after despreading with processing gain $G$, the *SINR* is below some threshold $\gamma$. In our simulations a threshold $\gamma = 10$ dB is used to determine if a node receives successfully.

The routing overhead as a function of network size is shown in Fig. 9. The routing overhead in LEACH is excessive and hinders the scalability. The reason is that LEACH assumes that all nodes are within communication range of each other and the sink. On the other hand, with DUCS the routing overhead is maintained well below 33% because the cluster-head advertisement messages are sent directly to the neighbours and not through the entire network.

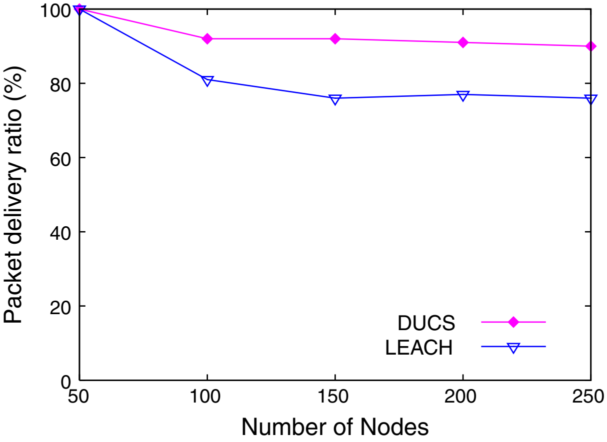


**Fig. 10** Packet delivery ratio as a function of network size

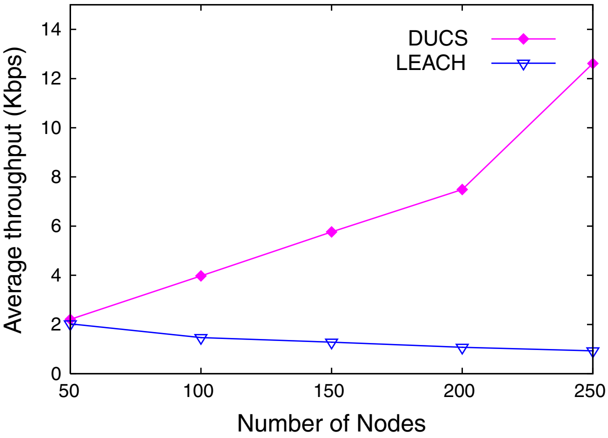


**Fig. 11** Throughput as a function of network size

Figure 10 shows the packet delivery ratio. With LEACH the packet delivery ratio is diminished with the network size, whereas DUCS achieves very high packet delivery ratios even in large network sizes because the use of timing advance and time guards enables to send properly more data packets and avoids acoustic collisions at the cluster head when cluster members using adjacent time slots send their data.

Figure 11 shows the average throughput. The average throughput is slightly reduced with the network size using LEACH to 0.9 Kbps for a network size of 250 nodes. The throughput is very low because LEACH is not well-suited for mobile underwater networks and the number of acoustic collisions at the sink is high. On the contrary, with DUCS the average throughput is increased with the network size to 12.6 Kbps for a network size of 250 nodes. DUCS improves the average throughput significantly in comparison to LEACH because more data packets are properly received adjusting the timing advance and time guard values to avoid acoustic collisions.

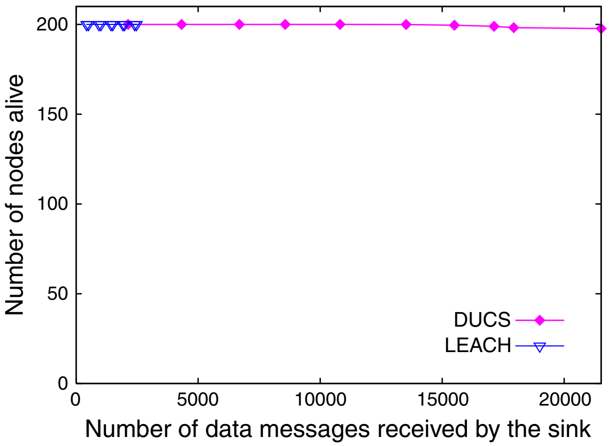


**Fig. 12** Number of nodes alive per data sent that arrives to the sink as a function of network size

Finally, Fig. 12 shows the number of nodes alive per data sent that arrives to the sink for $N = 200$ nodes. DUCS can deliver 8.7 times the amount of effective data to the sink as LEACH with four node deaths for the same simulation time.

## 7 Conclusions

In this paper we have explained the challenges involved in the design of new routing protocols suitable for UWSNs. Therefore, we propose DUCS as a new simple routing protocol specifically designed for long-term non-time-critical aquatic monitoring applications in underwater environments due to its fundamental properties: DUCS is simple, energy-aware and GPS-free; it minimizes the proactive routing exchange, uses data aggregation techniques and does not use flooding. Besides, DUCS assumes random node mobility and compensates the high propagation delays of the underwater medium using a continually adjusted timing advance combined with guard time values to minimize data loss. The combination of DUCS with TDMA/CDMA reduces interference and improves communication quality.

While DUCS appears to be a promising protocol, some principles in its design could be improved: The cluster members could only send information to the cluster head when they detect an interesting event and not continuously. Besides, the duration of the rounds has a significant impact on the overhead; although forming adaptive clusters is more energy efficient, the commitment between the energy saving and the overhead cost should be studied.

The analytical studies and simulations carried out demonstrate the scalability and effectiveness of the proposed scheme. DUCS achieves a very high packet delivery ratio while considerably reducing the network overhead and increasing the throughput; consequently, the basic characteristics of DUCS can be applied in the design of other routing protocols for UWSNs.

**Acknowledgments** This work was supported by the MEC (the Spanish Ministry of Education and Science) under the project TSI2007-66637-C02-01, which is partially funded by FEDER.

## References

1. Cui, J.-H., Kong, J., Gerla, M., & Zhou, S. (2006). Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. *Special Issue of IEEE Network on Wireless Sensor Networking, 20*(3), 12–18.
2. Makhija, D., Kumaraswamy, P., & Roy, R. P. (2006). Challenges and design of mac protocol for underwater acoustic sensor networks. In *4th WiOpt*, Boston, Massachusetts.
3. Xie, P., Cui, J.-H., & Lao, L. (2006). VBF: Vector-based forwarding protocol for underwater sensor networks. In *Proceedings of IFIP Networking*, Coimbra, Portugal.
4. Broecker, W., & Peng, T.-H. (1982). *Tracers in the sea*. New York: Eldigio Press.
5. Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks Journal (Elsevier), 3*(3), 257–279.
6. Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *ACM MOBICOM*, Boston, Massachusetts.
7. Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications, 1*(4), 660–670.
8. Lindsey, S., & Raghavendra, C. (2002). PEGASIS: Power-efficient gathering in sensor information systems. In *IEEE Aerospace Conference Proceedings* (Vol. 3, pp. 9–16).
9. Manjeshwar, A., & Agarwal, D. P. (2001). TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*.
10. Nicolaou, N., See, A., Xie, P., Cui, J.-H., & Maggiorini, D. (2007). Improving the robustness of location-based routing for underwater sensor networks. In *Proceedings of IEEE OCEANS'07*, Aberdeen, Scotland.
11. Pompili, D., Melodia, T., & Akyildiz, I. F. (2006). Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom)*, Los Angeles, CA.
12. Jornet, J. M., Stojanovic, M., & Zorzi, M. (2008). Focused beam routing protocol for underwater acoustic networks. In *WUWNet 2008*, San Francisco, CA.
13. Seah, W. K. G., & Tan, H.-X. (2006). Multipath virtual sink architecture for underwater sensor networks. In *Proceedings of the MTS/IEEE OCEANS2006 Asia Pacific Conference*, Singapore.
14. Lee, U., Kong, J., Magistretti, E., Park, J.-S., & Gerla, M. (2007). Time-critical underwater sensor diffusion with no proactive exchanges and negligible reactive floods. *Ad Hoc Networks Journal (Elsevier), 5*(6), 943–958.
15. Erol, M., Vieira, L. F. M., & Gerla, M. (2007). Localization with dive'n'rise (dnr) beacons for underwater acoustic sensor networks. In *Proceedings of WuWNet*, Montreal, QC.
16. Erol, M., Vieira, L. F. M., Caruso, A., Paparella, F., Gerla, M., & Oktug, S. (2008). Multi stage underwater sensor localization using mobile beacons. In *Sensorcomm, Second International Conference on Sensor Technologies and Applications*, Cap Esterel, France.
17. Stojanovic, M., & Freitag, L. (2006). Multichannel detection for wideband underwater acoustic CDMA communications. *IEEE Journal of Oceanic Engineering, 31*(3), 685–695.
18. Pompili, D., Melodia, T., & Akyildiz, I. F. (2007). A distributed CDMA medium access control for underwater acoustic sensor networks. In *Proceedings of Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Corfu, Greece.
19. Domingo, M. C., & Prior, R. (2008). Energy analysis of routing protocols for underwater wireless sensor networks. *Elsevier Computer Communications Journal, 31*(6), 1227–1238.
20. Salva-Garau, F., & Stojanovic, M. (2003). Multi-cluster protocol for ad hoc mobile underwater acoustic networks. In *IEEE OCEANS Conference*, San Diego, California.
21. Domingo, M. C., & Prior, R. (2007). A distributed clustering scheme for underwater wireless sensor networks. In *Proceedings of 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Athens, Greece.
22. Domingo, M. C., & Prior, R. (2007). Design and analysis of a GPS-free routing protocol for underwater wireless sensor networks in deep water. In *Proceedings of UNWAT* 2007, Valencia, Spain.
23. Mouly, M., & Pautet, M.-B. (1992). The GSM system for mobile communications. *Palaiseau*.
24. Rodoplu, V., & Kyoung Park, M. (2005). An energy-efficient mac protocol for underwater wireless acoustic networks. In *IEEE OCEANS Conference*.
25. Jourdain, G., Loubet, G., & Brossier, J. M. (1992). Underwater acoustical communication using pseudo-orthogonal codes. In *Proceedings of Underwater Defence Technology*, London.
26. Zhao, L., Hong, X., & Liang, Q. (2004). Energy-efficient self-organization for wireless sensor networks: A fully distributed approach. In *Proceedings of GLOBECOM '04*, Dallas, Texas.

27. Urick, R. J. (1983). *Principles of underwater sound*. Los Altos Hills: McGraw-Hill.
28. NS-2: Network simulator. http://www.isi.edu/nsnam/ns.
29. LinkQuest. http://www.link-quest.com.

## Author Biography

**Mari Carmen Domingo** received her Lic. degree in Telecommunications Engineering and her Ph.D. in Telematics Engineering from the Technical University of Catalonia, Barcelona, Spain in 1999 and 2005, respectively. She currently works as Lecturer at the Department of Telematics Engineering, doing research in wireless networks. She has published several papers in international journals and a book chapter. Dr. Domingo has worked on a number of national and European R&D projects. Her current research interests are in the area of mobile ad-hoc networks, wireless sensor networks, heterogeneous networks and distributed algorithms. She is an IEICE and IEEE member. She received the ALCATEL "Best Ph.D. thesis in wired-wireless convergence: applications and services" award from the Spanish Telecommunication Engineers Official Association (COIT) in 2006. From February 2008 to October 2008, she has been a postdoctoral researcher at the Broadband Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, Georgia, USA.