A DISTRIBUTED CLUSTERING SCHEME FOR UNDERWATER WIRELESS SENSOR NETWORKS

Mari Carmen Domingo
Technical University of Catalonia (UPC)
Barcelona, Spain

Rui Prior
University of Porto
Porto, Portugal

ABSTRACT

Underwater Wireless Sensor Networks (UWSNs) pose many challenges due to the intrinsic properties of underwater environments such as *large propagation delays*, *node mobility* and *limited bandwidth* capacity of acoustic channels. In this paper we present DUCS (Distributed Underwater Clustering Scheme), a new *GPS-free* routing protocol for UWSNs 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 and maintain communication quality. The simulations carried out demonstrate the effectiveness of the proposed scheme.

I. INTRODUCTION

During 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 the 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 oil spill) and the existing technologies do not measure up to the demanding requirements. Consequently, a new concept of *low-cost easier deployable* underwater networks with less restricted conditions should be developed: Underwater Wireless Sensor Networks (UWSNs) [1]. This kind of networks should be *scalable*, *mobile* and capable of *self-organization.* They eliminate the need for cables and are based on the propagation of acoustic waves [2].

UWSNs are a new research paradigm that poses exciting challenges due to the intrinsic properties of the underwater environments. They suffer:

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

Thus, the restricted network operation conditions pose a motivation for doing research at each layer of the 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 can not 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* support. We have validated the efficiency of our routing protocol through simulations.

The paper is structured as follows. Section II explains related work about the design of routing protocols in UWSNs. Section III describes in detail the functioning of our routing protocol. Section IV shows our simulation results. Finally, Section V concludes this paper.

II. RELATED WORK

Some routing protocols [3] [6] have been specifically designed for UWSNs. As common characteristics they assume GPS-free nodes, random node mobility and no proactive design. However, in the proposed geographic routing protocol [3] some nodes along a routing vector always forward packets from concrete sources to the sinks and their battery capacity can be quickly exhausted, especially if the forwarding paths intersect each other in the case of multiple sources and sinks. It would be recommended to design a protocol where intermediate nodes are changed along the time. On the other hand, the proposed routing protocol in [6] uses 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 is considered a superior protocol for underwater environments due to its properties and we think that the research efforts should

This work was supported by the MEC (the Spanish Ministry of Education and Science) under the project TSI2006-13380-C02-01, which is partially funded by FEDER.

1-4244-1144-0/07/$25.00 ©2007 IEEE.

concentrate on techniques to improve its limitations (e.g. spatial reuse of the spreading codes).

In [7] the authors propose another clustering protocol that uses TDMA/CDMA for network communication. However, this solution is restricted because it assumes that cluster formation and maintenance is based on the nodes position and movement information received using cables and GPS, whereas DUCS is 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 and the timing advance technique to compensate high underwater propagation delays.

III. DUCS Protocol

A. *Protocol Architecture*

DUCS is an *adaptive self-organizing* protocol where clusters are formed using a *distributed* algorithm. We suppose that underwater sensor nodes always have data to be sent to the sink and that they can use *power control* to adjust its transmission power.

In DUCS the nodes organize themselves into local clusters and one node is selected as 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. aggregation) and transmits the data to the sink using multi-hop routing (relaying it through other cluster-heads). Nodes close to each other process very frequently correlated data because they monitor the same phenomena and with the aid of aggregation techniques the effective non-redundant data can be extracted by the cluster-head and send to the sink. Thus, energy is saved. Cluster-heads are responsible for coordination among nodes within their clusters (intra-cluster coordination) and communication between each other (inter-cluster communication).

DUCS incorporates randomized rotation of the cluster-head among the sensors to avoid fast draining of the batteries of specific underwater sensors. In this way, the energy consumption is distributed. The functioning operation of DUCS is divided into rounds (see Fig. 1). Clusters are formed during the set-up or clustering creation process and data transfer occurs during the network operation or steady-state 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 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).

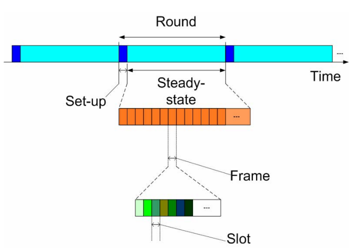


Fig. 1. Time line of DUCS.

Both phases are repeated periodically. The network operation phase is long compared to the clustering creation process to minimize the overhead.

B. *Cluster-head Selection Algorithm*

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

$$CH_{prob} = \frac{C_i}{C_{MAX}} \times C_{prob} \tag{1}$$

, where C_i represents the node's battery level (residual energy of the node) and C_{MAX} represents the maximum battery capacity. C_{prob} is a small constant fraction used to set an initial percentage of cluster-heads limiting the number of initial cluster-head announcements. CH_{prob} is not allowed to fall below a small probability, $p_{\min}$; this restriction is necessary to increase the probability that some nodes elect themselves as cluster-heads even if the sensor battery levels in the whole network are scarce to ensure that the routing protocol will still function.

Each non-cluster head decides to which cluster it belongs by choosing the cluster-head that requires the *minimum communication energy* and consequently its *power level* required for *transmission* is *minimized*. The transmission power is directly proportional to the distance between sensor nodes in shallow water scenarios (with sea depth lower than 100 *m*). 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.

During the clustering creation process, the nodes compute their remaining energy and calculate their probability to become a cluster-head CH_{prob}. If CH_{prob} falls above a random value between 0 and 1, a node elects itself as cluster-head and sends an advertisement message to its neighbours using CDMA.

During this phase, a node processes the cluster-head announcements it has received to select the lowest cost cluster-head. After each node has decided to which cluster it wants to belong, it must inform the cluster-head sending a join-request message back using CDMA. If a node has not received any cluster-head announcements, it should decide to send its data directly to the sink.

As a result, nodes with higher residual energy become cluster-heads and lower intra-communication cost is spent.

C. *Cluster Formation Algorithm and Network Operation Phase*

After the clustering creation process is over, each 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 (Code Division Multiple Access) to the cluster members. 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 communication loss. The solution proposed in this paper 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) [8]. The timing advance value for each node can be computed only by the cluster-head, and is then provided 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 CH_j sends acoustic signals to know the propagation delays from itself to each cluster member, which are τ_1, τ_2, $\tau_3 \ldots \tau_f$ respectively.

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 τ_l; therefore it establishes (once the schedules have been sent) a starting moment for transmission after τ_l; thus all nodes adjust the same reference starting moment.

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.

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. 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 τ_s seconds (lowest propagation delay) the transmission of the first frame and from here on frames are sent uninterruptedly. This means that in the schedule the node N_s with the minimum propagation delay τ_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:

$$T_{start_{s+1}} = \tau_s + T_{slot_s} - \tau_{s+1} \qquad (2)$$

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

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

$$T_{start_{s+i}} = \tau_s + \sum_{j=s}^{s+i-1} T_{slot_j} - \tau_{s+i} \qquad (3)$$

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:

$$T_{start_{s+i}} = \tau_s + t \times \sum_{j=s}^{s+u} T_{slot_j} + \sum_{j=s}^{s+i-1} T_{slot_j} - \tau_{s+i} \qquad (4)$$

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

We suppose that always $T_{start_{s+i}} \geq 0$.

In Fig. 2 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.

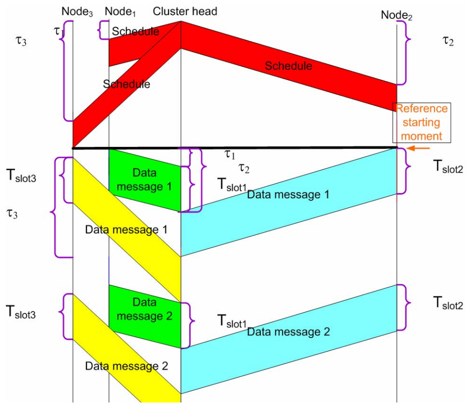


Fig. 2. Example of a time diagram.

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 [9]. Therefore, in order to avoid collisions between two non-cluster head members whose data messages arrive simultaneously to the same cluster-head, 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 is set to one tenth of a time slot.

After the transmission of a frame to the cluster-head, a 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 functioning of the cluster maintenance algorithm is the following one:

During the time slot reserved for maintenance purposes each node should again estimate its propagation delay to its cluster-head in its cluster with the aid of ToA techniques (each cluster-head sends a broadcast packet periodically to maintain the synchronization required in TDMA). If this delay is β% 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 afterwards it should select the cluster-head neighbour that is closer to it. 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. β is set be 50%.

When the cluster-head has received a data message from each cluster member, it compresses 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. Each cluster-head CH_i selects as next hop to forward its frames towards the sink S the adjacent cluster-head CH_j (that minimizes the distance $CH_i - CH_j$) and satisfies: distance $CH_i - CH_j <$ distance $CH_i - S$ and distance $CH_j - S <$ distance $CH_i - S$. CH_i knows the distance $CH_j - S$ because the advertisement message sent by CH_j during the set-up phase includes this information.

D. *MAC protocols*

We propose to use TDMA and CDMA with DSSS (Direct Sequence Spread Spectrum) using pseudo-orthogonal codes [10] for intra-cluster communication and only CDMA with DSSS using pseudo-orthogonal codes in all other communications processes. With DUCS each node that elects itself as cluster-head, selects randomly a unique spreading code to send an advertisement message to their neighbours. If a node receives two advertisement messages from two different cluster-heads using the using the same spreading code, it should advertise one of them so that this cluster-head should send again a new advertisement message with a new different spreading code that invalidates the previous one. If there are no more different spreading codes available, this node can not declare itself as cluster-head. 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 an advertisement message, 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 CDMA with the same spreading code and again the same code is used by the cluster-head when it sends the aggregated data to the next hop towards the sink using multi-hop routing. Fig. 3 and Fig. 4 illustrate an example of code assignment.

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 code for transmission.

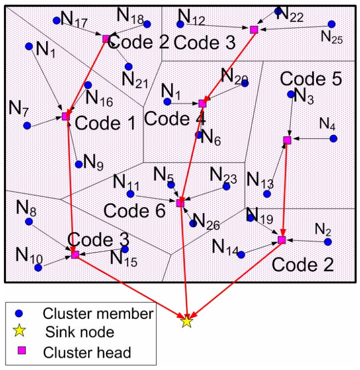


Fig. 3. Example network with DUCS.

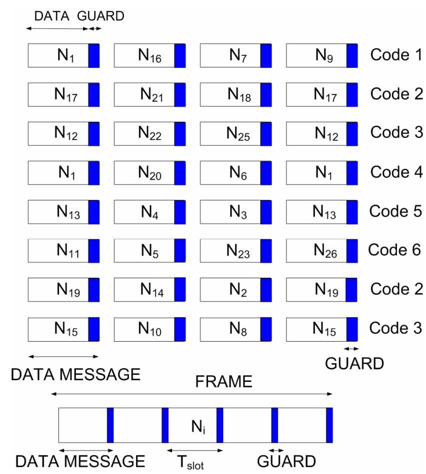


Fig. 4. Frame structure and code assignment.

IV. Simulations

We have run simulations with the NS-2 [11] tool to investigate the performance of our proposed approach. Therefore, we have modified the physical and MAC layers to support underwater communications. The chosen scenario consists of N sensor nodes that communicate with a sink. The underwater sensor nodes are uniformly distributed in a volume of $S \times S \times H$ 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 $75 \times 75 \times 75$ 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 [12].

We have run 30 simulations for 1000 seconds to compare the performance of LEACH [13] with our proposed routing protocol in terms of packet delivery ratio (percentage of data packets successfully delivered), average routing overhead and number of nodes alive per amount of data messages sent that arrive to the sink. In LEACH the number of clusters per round is set to be *N/5*. The duration of a round for both protocols is set to 200 seconds. In our simulations the DSSS system has a 5 Megachips per second code clock rate, so that the processing gain $G = 714.3$. An outage event occurs when, after despreading with G, the $SINR$ is below some threshold γ. 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. 5 (a). 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 30% because the cluster-head advertisement messages are sent directly to the neighbours and not through the entire network.

Fig. 5 (b) 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.

Finally, Fig. 5 (c) shows the number of nodes alive per data sent that arrives to the sink for $N = 200$ nodes. DUCS can deliver eight times the amount of effective data to the sink as LEACH with four node deaths for the same simulation time.

V. Conclusions

In this paper 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. The 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.

References

[1] J.-H. Cui, J. Kong, M. Gerla and S. Zhou, "Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications", *Special issue of IEEE Network on Wireless Sensor Networking*, May 2006.

[2] M. Stojanovic, "Acoustic (Underwater) Communications", *entry in Encyclopedia of Telecommunications*, John G. Proakis, Ed., John Wiley & Sons, 2003.

[3] P. Xie, J.-H. Cui and L. Li, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks", *UCONN CSE Technical Report: UbiNet-TR05-03*, Feb. 2005.

[4] W. Broecker and T.-H. Peng, "Tracers in the sea", *Eldigio Press, Lamont Doherty Earth Observatory of Columbia University*, Palisades, NY, page 689, 1982.

[5] I. F. Akyildiz, D. Pompili and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges", *Ad Hoc Networks Journal, (Elsevier)*, Mar. 2005.

[6] U. Lee, J. Kong, E. Magistretti, J.-S. Park and M. Gerla, "Time-critical Underwater Sensor Diffusion with No Proactive Exchanges and Negligible Reactive Floods", *Ad Hoc Networks Journal (Elsevier)*, 2007.

[7] F. Salva-Garau and M. Stojanovic, "Multi-Cluster Protocol for Ad Hoc Mobile Underwater Acoustic Networks", *In IEEE OCEANS'03 Conference*, 2003.

[8] M. Mouly, M.-B. Pautet, "The GSM System for Mobile Communications", *Palaiseau,* 1992.

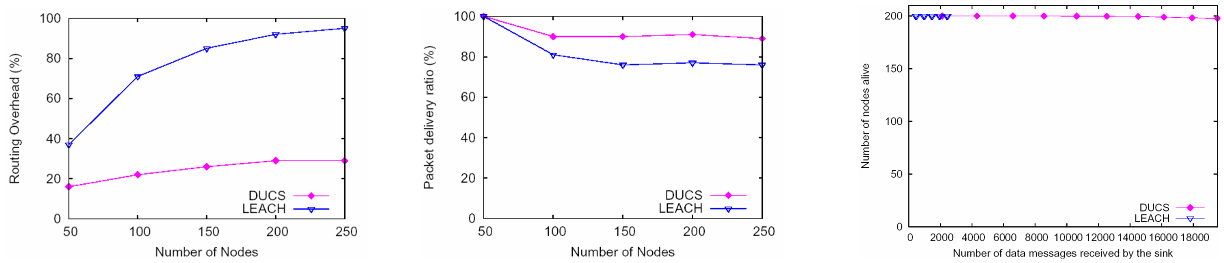
[9] V. Rodoplu and M. Kyoung Park, "An Energy-Efficient Mac Protocol for Underwater Wireless Acoustic Networks", *IEEE OCEANS'05 Conference*, Sept. 2005.

[10] G. Jourdain, G. Loubet, "Underwater acoustical communication using pseudo-orthogonal codes", *Underwater Defence Technology*, 1992, London.

[11] NS-2: Network Simulator, http://www.isi.edu/nsnam/ns.

[12] LinkQuest, http://www.link-quest.com.

[13] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Trans. on Wireless Communications, Vol. 1, No. 4, Oct. 2002, pp. 660-670.



(a) Average routing overhead (b) Average packet delivery ratio (c) Number of nodes alive per data sent

Fig. 5. Performance comparison between LEACH and DUCS.