A Mobile Delay-tolerant Approach to Long-term Energy-efficient Underwater Sensor Networking

Eugenio Magistretti
DEIS
University of Bologna
emagistretti@deis.unibo.it

Jiejun Kong, Uichin Lee, Mario Gerla
Computer Science Department
UCLA
{jkong,uclee,gerla}@cs.ucla.edu

Paolo Bellavista, Antonio Corradi
DEIS
University of Bologna
{pbellavista,acorradi}@deis.unibo.it

***Abstract*— Underwater environment represents a challenging and promising application scenario for sensor networks. Due to hard constraints imposed by acoustic communications and to high power consumption of acoustic modems, in Underwater Sensor Networks (USN) energy saving becomes even more critical than in traditional sensor networks. In this paper we propose Delay-tolerant Data Dolphin (DDD), an approach to apply delay-tolerant networking in the resource-constrained underwater environment. DDD exploits the mobility of a small number of capable collector nodes (namely dolphins) to harvest information sensed by low power sensor devices, while saving sensor battery power. DDD avoids energy-expensive multi-hop relaying by requiring sensors to perform only one-hop transmissions when a dolphin is within their transmission range. The paper presents simulation results to evaluate the effectiveness of randomly moving dolphins for data collection and paves the way for future analytic investigations.**

I. Introduction

The vastness of the ocean, covering about two-third of the surface of earth, is still largely unexplored. This calls for the need of implementing unmanned underwater sensor network (USN) for periodic oceanographic monitoring. The new USN paradigm, however, poses formidable new challenges. Let us use the existing wireless radio sensor networks for a comparison. In contrast to wired networks, wireless radio networks operate in a resource constrained environment. Based upon technology for Dense Wave Division Multiplexing (DWDM), a single optical bundle can carry 12,800 GHz of optical signal. However, even the richest frequency band owner in the United States, namely Department of Defense (DoD), only owns approximately 300MHz of the total 3GHz of useable radio spectrum. Of the 300MHz owned by the military, individual systems are allocated in small blocks, e.g. 10MHz, 1MHz, or less. Consequently, protocols in wireless radio networking must be far more efficient than the same protocols in wired networks.

Nevertheless, if we extend our comparison to the underwater world, then wireless radio networks become the one with relatively much richer resource to expend. As high-frequency radio signals cannot propagate in water, underwater networking must rely on low-frequency acoustic communication, with the frequency upper bound reported as 1MHz at 60-meter range and reduced to the KHz level as the transmission range increases [12]. This implies that the entire acoustic band is less than several MHz and typical allocation is measured in KHz for individual systems. This drastic reduction in communication resource makes underwater networking an extremely challenging topic. The underwater wireless acoustic networks demand several orders of magnitude improvement in protocol efficiency compared to their "resource-abundant" wireless radio counterparts. In addition, acoustic communications impose far higher power consumption than RF based. The mechanical generation of acoustic waves, e.g., through ceramic projectors [6], makes modems power-hungry. Just to give some examples, the RF transmitter embedded in Crossbow Mica2 Motes [15], used for ground-based wireless sensor networks, consumes around 50mW to cover up to several tens of meters. State-of-the-art acoustic modems [6] [14] for USN drains 2 Watts to reach one to two-hundred meters.

In this paper we explore the power of mobility to alleviate the problems caused by the extremely constrained underwater communication. Based on the promising paradigm of delay-tolerant networking [5], we propose the design of *Delay-tolerant Data Dolphin* (DDD) to maximize the network lifetime of USN. DDD exploits mobile collectors, i.e., dolphins, scouring a region of interest where stationary sensors have been formerly deployed. Each underwater sensor is only required to transmit its data reports to the nearest dolphin in short one-hop distance, when the dolphin reaches its neighborhood. We have mainly verified our design by simulation studies, investigating how parameter setting influences the effectiveness and efficiency of DDD. Since we consider the number of dolphins as a crucial design parameter, we have explored how the collection accuracy and the delay needed to harvest the reports depend on it. Finally, we observe that the problem configures for a possible stochastic investigation. In the last part of this contribution we explore the single-dolphin case by proving that, in a one-dimensional approximation, the time needed for a dolphin to cover the whole area is proportional to the square of the number of states.

The paper is organized as follows. Section II presents an overview of related work about delay-tolerant networking and underwater communications. Section III introduces DDD system architecture and protocol design. Then, Section IV presents our simulation results, while Section V shows our current analytic results. Section VI concludes the paper.

II. Related Work

DDD topic relates mainly to two different research areas, i.e., delay-tolerant and underwater networking. In the following, we discuss state-of-the-art contributions in these fields.

A. Delay-tolerant networking

Many types of applications and challenging environments call for the introduction of the delay-tolerant architecture, a novel networking paradigm capable of operating in situations where simultaneous presence of senders and receivers cannot be assumed. Typical examples are exotic networks, e.g., deep space links [2], military networks, and underwater networks, where the limited transmission range and the mobility of the communication subjects prevent the establishment of persistent links.

These environments introduce critical aspects, hardly challenging traditional Internet architectures. In [5], Fall identifies a set of peculiar elements characterizing these types of networks: first of all disconnections, induced by mobility and energy conserving sleep states; in addition, high latency paths and low allowed data rates (due to radio and acoustic media). From a system point of view, interoperability and security represent critical challenges.

In order to face these challenges, a foundational architecture is proposed, called Delay Tolerant Networking (DTN) [5]. This aggregates heterogeneous networks, called *regions*, through DTN *gateways*. DTN gateways are in charge of moving messages from one of their adjacent regions to another, by physically transferring the information and by performing needed translations. Many challenges are identified, e.g., message routing, delivery reliability, retransmission management, security and authenticity, and flow/congestion control.

Research instances of delay-tolerant networking architectures have been proposed in the last years. [20] investigates epidemic protocols for the delivery of information about whales behavior to *Infostations*. Whales are equipped with sensing and communication devices. As soon as two whales are within mutual transmission ranges, they exchange carried data with a certain probability. When one whale reaches an *Infostation*, it offloads all gathered data. Differently from our proposal, [20] does not consider mobile collectors, but mobile sensing devices that epidemically spread information through other participants.

Another interesting example is provided in [23], where sparse networks are supported by *message ferries*. Remote node communities, without any chance of direct Internet connection, dispatch messages to ferries periodically passing in their neighborhood. Ferries provide communication services to the nodes, by delivering their messages to the Internet. Ferries are able to plan and modify their routes according to end user needs. Let us observe that in [23] remote communities are formed by mobile nodes and that ferries routes are not random, but can be planned proactively or reactively.

B. Underwater Networking

Some pioneering works on USN have been recently proposed. Some tutorial articles identify the main research challenges in this novel and uncharted area [9] [1] [4], by mainly concentrating on hardware and protocol issues. As deeply discussed in [17], underwater environment makes radio communications infeasible, due to the exceedingly high absorption rate, and calls for the deployment of acoustic modems. These devices are energy consuming, permit only low data rates, and impose an extremely high communication delay due to the low propagation speed of the acoustic waves. This effect highly challenges protocol design, and in particular medium access control solutions [17] [9] [4].

Other crucial services, like time synchronization and localization [9], call for novel research contributions to deal with the harsh environment. Networking and transport layers are not exempt from these issues [1]. A cross-layer design approach, including power management, coordination, and localization planes is highly advocated in [1]. [9] envisions interesting applications for untethered sensors deployed in the sea. [4] presents the interesting paradigm of mobile USN, i.e., networks where the underwater sensors move conveyed by water currents. Two interesting architecures are investigated, the one facing long-term non-time-critical applications (e.g., oceanography and pollution detection), the other short-term time-critical explorations (e.g., resource discovery and military surveillance).

Despite this significant work envisioning novel applications and high-level architectures, not much research has addressed the design of supporting communication protocols. Most original contributions have focused on acoustic modems design [6]. We propose DDD as one of the first efforts in the investigation of communication protocols specific for this uncharted field.

III. Delay-tolerant Data Dolphins

Many promising underwater applications such as seismic monitoring for oil extraction [9], ocean sampling and undersea exploration [1] call for the deployment of a large number of small stationary sensors. Then, sensed data can be collected in different ways: the traditional approach [17] is to recover the sensors after the task has ended; or alternatively, sensors can directly transmit data to remote stations (e.g., buoys). Both these solutions suffer from disadvantages. In the first case, time for data harvesting is forced to be the end of network lifetime. The system is inherently much more prone to unexpected inconsistencies or even collapses (due to lack of reconfiguration capabilities). In addition, regular nodes have to store the whole sensed data until they are recovered. In the second case, reports reach remote stations much faster, but they likely need to travel long paths, thus draining energy of nodes en route. Long-range communication also increases channel interference amongst different transmitters, in particular in the acoustic channel with very long propagation delay.

In this paper we study a delay-tolerant approach that permits to trade delay for energy consumption. DDD inherits advantages of both previously described approaches and, for instance, suits the periodic oceanographic monitoring applications, which can typically collect data periodically.

A. Architecture

In DDD, small stationary sensor devices are deployed on the seabed and continuously collect data from the environment. Data is locally processed, and is stored only upon the occurrence of interesting events (i.e., of special events that are designated by the command center). Beside sensors, a number of moving vehicles, namely *dolphins*[1], equipped with more capable devices scour the ground field to harvest data scattered over the network of sensors. Dolphins receive data packets only when they move within the one-hop neighborhood of related sensor nodes. Then, dolphins deliver gathered events as soon as they reach a remote base station.

This strategy permits sensors to spare the energy needed for long-range communications toward remote stations, at the price of a delayed delivery. Let us observe that considered applications are not time critical, thus do not impose real-time constraints on data delivery. However, compared to traditional solutions based on sensor recovery at the end of network lifetime, data harvesting delay is highly reduced. Simulation results will show that DDD system parameters, e.g., number of dolphins and buffer size, can be tuned to fit the delay requirements of the single application.

Sensor node assumptions. Sensors are energy and memory limited. To save energy, they spend most of time in a low-power sleep state, which reduces energy expense in various components like the sensor acoustic modem, the sensor sensing unit and the sensor mainboard. A sensor only periodically wakes up for environment sensing and event generation. Its communication component is an acoustic modem that includes two components [22]. One component mostly operates acoustic communications with near dolphin nodes. The other is a low-power transceiver used to determine the presence of dolphin nodes (by a special signal transmitted from dolphin nodes) and trigger the first one. Thus, the latter is turned on only when an event needs to be delivered, the former only when the first determines the presence of a dolphin. Sensors are equipped with a small amount of memory, so that they store a few recently-detected events. Due to the memory constraint, events need to be replaced as memory fulfills.

In many applications, sensor data report must be tagged with location and time. We assume that each stationary sensor already knows its position. This can be the result of a careful deployment practice, or of the execution of a sensor localization protocol, which is a research challenge studied in many research projects [9] for the harsh underwater environment.

Dolphin node assumptions. On the other hand, dolphin nodes are supposed to have large memory and energy conserves if compared with sensors. They collect and store events while moving in the field, and upload their memory to a near-shore base station for data delivery. Hence, they need to bear all events harvested between two consecutive passages to the base station (or a set of base stations in distributed deployment). In addition, they are equipped with powerful batteries, to overcome limitations on costly acoustic communication.

Dolphin nodes can move with random or controlled mobility. Since we cannot foresee where and when an event will be generated, trajectory planning would be hardly effective for a single dolphin node. Therefore, we study the random mobility case where dolphin nodes move by following the random walk model. Every dolphin node moves with an independently and identically distributed random pattern [13]. Our design architecture could permit also to exploit controlled mobility, where dolphin nodes move on a determined route computed from Hamilton route problem [8]. The runtime investigation of an approximated polynomial-time solution to this NP-complete problem would be costly and is beyond the scope of this paper.

B. Protocol Design

To save energy, we propose that sensors exploit one-hop communications toward dolphins, i.e., that sensors deliver events only when a dolphin is within their sensing range. Once delivered, these events are stored in dolphin memory and can be safely removed from sensor memory.

Dolphins randomly move in the monitored area A, by periodically broadcasting beacons to advertise their presence. Beacons need to be transmitted on acoustic frequencies compatible with the low-power modems of the sensors. Advertising periods t are chosen according to sensor deployment and communication range r, and to the speed of the dolphins v. In this paper, we choose $t << \frac{r}{v}$ so that we can guarantee that almost every time a dolphin is within the communication range of a sensor, the sensor can realize its presence. This choice will drain more power from dolphin battery but differently from sensors, this has a much longer lifetime and can be easily replaced.

As soon as they determine the presence of a dolphin, sensors turn on acoustic modems and upload stored events to the dolphin. Since dolphin speed is low (on the order of few m/s for underwater motion), every time a sensor determines the presence of a dolphin, it will be able to deliver all stored events. Then, when dolphin moves in the communication range of a base station, it uploads all gathered events.

Dolphins will likely be costly nodes. As a consequence, our goal is to reduce the number of dolphins needed while serving the aquatic application requirements. If the number of deployed dolphins is not large enough, they will not be able to collect all needed event reports generated by the deployed sensors. Thus, our final goal is to determine the minimal number of dolphins that permits to statistically collect the requested large share of events. Since dolphins perform random motions and events are removed from sensor buffers when there is no space left, we do not expect to be always able to exhaustively collect all events. Instead, we want to investigate the probabilistic relation between the number of dolphins and the delivery ratio of sensor event report.

[1]Let us remark that we do not just consider dolphins to carry the task, but any kind of fish sufficiently big to be equipped with a one-ounce device.

IV. Simulation Results

As a first effort in delay-tolerant networking for underwater scenarios, we simulated DDD on the well-known Qualnet simulator [18]. We modified Qualnet as in [21] to simulate underwater environments. We implemented a new medium and propagation model for underwater. In particular, the propagation speed of the communication signal was set to the speed of sound in water, i.e. *1500 m/s*. In addition, we used a bandwidth of 20 Kbps and a channel frequency of 50 KHz according to the values found in [12].

As a deployment scenario we considered a *2000m x 2000m* area, with 25 sensor devices, uniformly deployed on a *5x5* grid spanning the whole square (these measures are chosen according to the ranges given in [4]). We decided to model the mobility pattern of dolphins according to a general mathematical model, the random walk [13] [16], essentially based on brownian motion. Random walk logically splits time in identical slots; at the beginning of each slot, each dolphin chooses a constant random direction and follows that direction for the duration of the whole slot with a constant speed equal to *5m/s*. We realize that this assumption could be considered oversimplifying, but actually some recent research on fish movement presents similar models [11].

Let us observe that our goal is not to present precise results for the single examined application, but rather to show performance trends. We claim that by stretching or shrinking the deployment area, and varying dolphin speed or movement patterns, general trends will not be affected.

The results presented in the following refer to scenarios where sensors generate new events according to a Poisson distribution with average value *lambda*. Sensors store generated events in a local buffer with size *buf size*, until (a) the buffer is full (in that case we adopt a "least recently generated" replacement policy), or (b) the events have been uploaded to a dolphin (in that case, the buffer is emptied). Dolphins are equipped with an unbounded buffer. Where not differently specified, default values for *lambda* and *buf size* are chosen equal respectively to 1 event every *2 minutes* and *100* entries.

The goal of our simulations was to assess how the harvesting *Accuracy* and *Delay* depend on three principal system parameters, i.e., the number of dolphins, the generation rate *lambda*, and the buffer size *buf size*. In more details, we define *Accuracy* as the total number of events collected by the dolphins (i.e., the cardinality of the union of the events harvested by any single dolphin), out of the number of events generated in the area (i.e., the gross total of the number of events generated by any single sensor). The *Delay* is instead defined as the average delay intercurring between the time an event is generated and the time it is harvested for the first time by any dolphin (a similar metric is considered for the evaluation of packet delivery delay in MANET routing protocols [3]). All values presented in the following figures are the average of twenty independent simulation runnings. Obtained results showed a really low variance.

First, we investigate how the *Accuracy* depends on the number of deployed dolphins and on the *buf size*.

Figure 1 shows the *Accuracy* as a function of the number of dolphins (the plots refer to extreme *buf size* ranging from 1 to 500). As we expected, increasing the number of dolphins has a significant impact at first. As soon as a threshold value has been reached, the contribution of additional dolphins does not carry appreciable benefits. Let us rapidly note that the threshold value highly depends on event generation rate and buffer capacity of deployed sensors. As pointed out in Section III, DDD design does not address applications where an exhaustive collection of events is of vital importance. As a consequence, the number of dolphins suitable for the single case depends on application requirements.

Figure 2 investigates how the *Accuracy* depends on the size of local buffers (the plots refer to 1, 3 and 7 dolphins cases). The growth is steeper in the first part of the graph, i.e., as the *buf size* increases from 1 to 100, then it slackens. This becomes evident as the number of dolphins increases: for the 7-dolphin case enlarging the buffer from 1 to 100 (resp. from 100 to 500) leads to a 56% increase (resp. 20%). In fact, by enlarging the *buf size* from 100 to 100+*x*, it is likely that, due to the high *Accuracy* of the multiple-dolphin case, most of the *x* additional entries harvested by a dolphin while in touch with a sensor have previously been collected by other dolphins, thus carrying no benefits.

Figure 3 shows how the generation rate *lambda* impacts on the *Accuracy*. The plots show results for *lambda* ranging from 1 event every minute to 1 every 200 minutes, 1 and 7 dolphins, and *buf size* equal to 1 and 100. As expected, as the generation rate decreases (i.e., as *lambda* decreases) the percentage of harvested events increases. As the plot for 7 dolphins and *buf size* equal to 100 clearly shows, there exists a lower bound after which there is no substantial *Accuracy* gain. When a certain parameter setting leads to an *Accuracy* approaching 1, the only events that are not collected at any moment are the ones recently generated. In fact, it is not feasible to obtain an *Accuracy* equal to 1 because, as soon as dolphins move in a region, new events could be generated in previously visited areas. The number of these events has a very small influence on the gross total and mainly depends on *lambda* and dolphin speed.

Figures 4 and 5 show how the *Delay* depends on the number of dolphins and on the *buf size* for two extreme values of *lambda*, 1 event every resp. 2 and 200 minutes. Both figures present the *normalized Delay*, i.e., the ratio between the *Delay* and the generation rate *lambda*. As the number of dolphins increases, the *normalized Delay* decreases, since events can be harvested more promptly by deploying a higher number of dolphins. As the *buf size* increases, the average harvesting delay increases. In fact, growing the *buf size* helps delivering to the dolphins more events, which have been generated less recently. This contributes improving the *Accuracy*, but on the other hand, introduces older elements in the calculation of the *Delay*. Figure 5 shows that this holds until a threshold value is reached (dependent on the generation rate *lambda*). Let us consider the plots corresponding to *buf size* equal to 10 and 100. Due to the low generation rate, sensors are visited with a frequency that impedes they fulfill their buffer between two subsequent visits. Thus, the *normalized Delay* tends to converge to the values of the plots corresponding to smaller *buf size*.

A crucial performance indicator is the time the dolphins need to harvest the events generated by a sensor. Obviously, since dolphins move randomly, the highest possible delay needed to get in touch with a chosen sensor is unbounded. However, we can experimentally determine an average measure of that interval. To this end, we set up a simulation where each sensor generated an event at time $T = 0$ and we measured the time needed to collect the events depending on the number of dolphins. Figure 6 reports on the y-axis the ratio of the sensors (out of the 25 deployed) visited after a certain time by the dolphins. Plots are averaged over 100 simulations and refer to the cases of 1, 3 and 7 dolphins. We observe that this final plot does not add qualitative information to the other figures, but only provides a tool for network designers to choose the number of dolphins they need to deploy depending on application-specific deadline requirements.

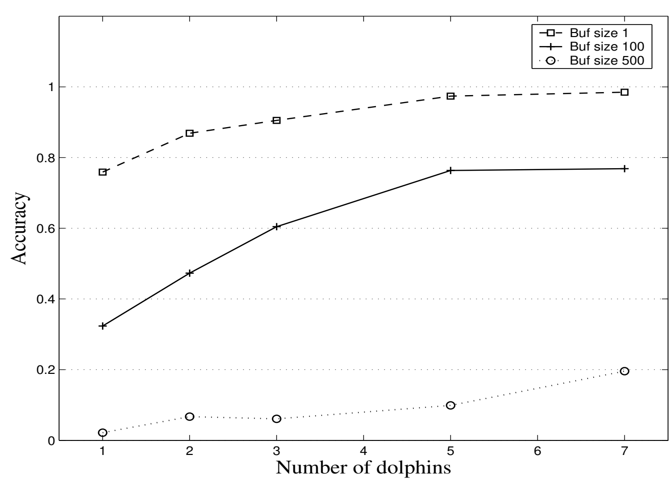

Fig. 1. Accuracy as a function of number of dolphins

V. Analysis

The previous section examines *Accuracy* and *Delay* as the main performance indicators of the system. As a matter of facts, it is rather difficult to analytically model and prove the trends of those plots. Our research efforts aim to determine another interesting indicator that tightly represents a gross value of the *Delay*, i.e., the time needed for one agent to gather all events generated in the network. First, we show a general problem formulation and underlying model. Then, we consider a one-dimensional analysis, by assuming random walk, as dolphin mobility model.

A. Underlying Model and Problem Formulation

Let us consider a sensor network composed of *n* sensors randomly scattered in a square area A. Sensors have a radio

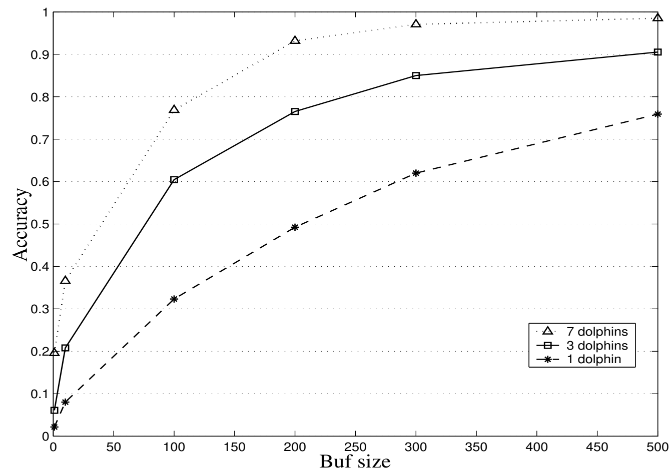

Fig. 2. Accuracy as a function of Buf size

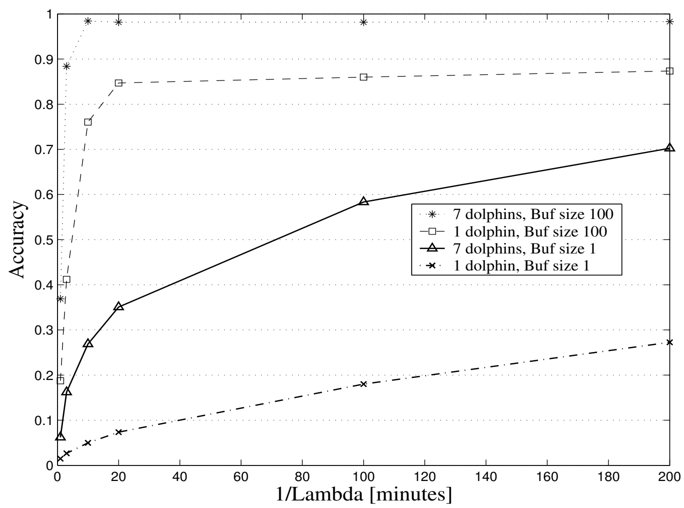

Fig. 3. Accuracy as a function of the generation rate

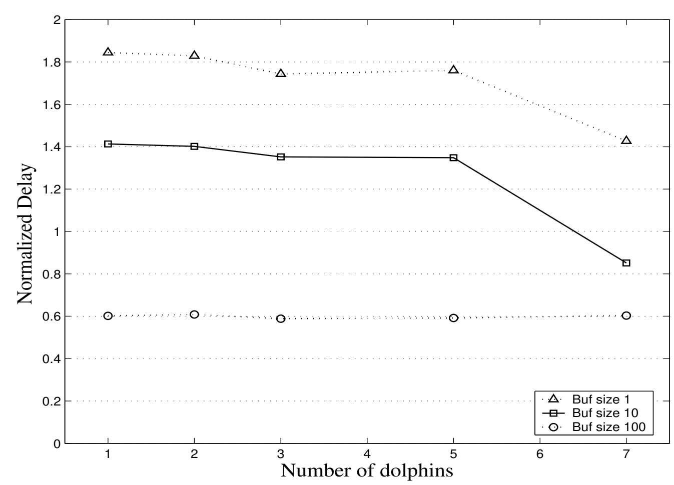

Fig. 4. Normalized delay for lambda=1 event every 2 minutes

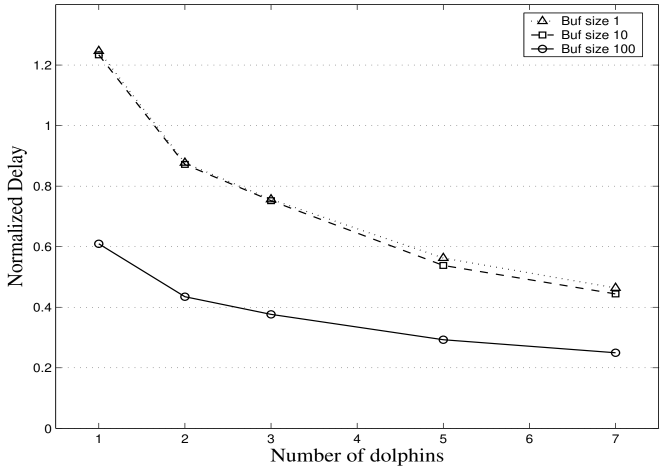

Fig. 5. Normalized delay for lambda=1 event every 200 minutes

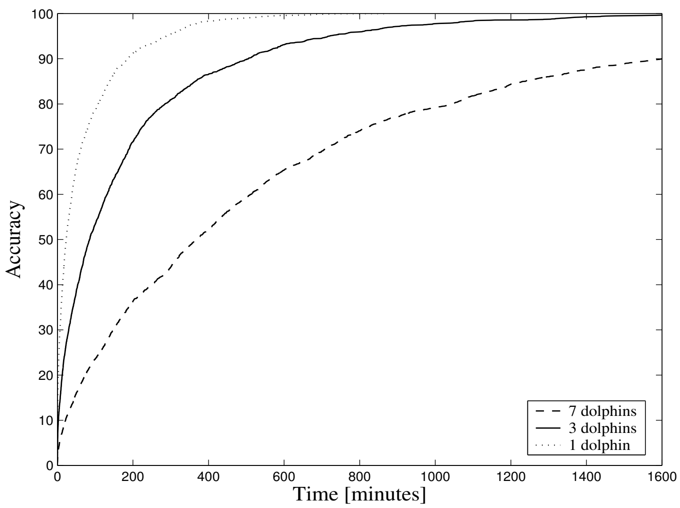

Fig. 6. Time evolution of Accuracy as a function of the number of dolphins

communication range equal to r. Every sensor senses a new event according to a Poisson distribution with expected value λ. In other words, each sensor generates new data reporting the sensed event according to a Poisson distribution independently from all the others.

The data generated from the sensors are to be collected by k dolphins floating within the area A. Dolphins are equipped with a communication device with range r and independently move in the deployment area A, without following any fixed or predetermined pattern. We assume each one performs independent random walks bounded by the area A, with a constant speed v. This means that every time period T, each dolphin chooses a random direction and moves in that direction for the next T instants with speed v. The data generated by sensors should be delivered to a dolphin as soon as possible. On the other hand, as motivated above, energy saving issues impose that data travel as few hops as possible, thus precluding or at least discouraging the use of multihop solutions. For this reason, each sensor delivers the gathered data only when a dolphin is within its communication range.

Due both to the limited memory of these small sensor devices and to the transient nature of the data, we suppose that events are to be delivered within a certain period. This data expiration interval could depend on the buffer size *buf size* of the sensors or be established for each event according to a Poisson distribution with expected value μ. If no dolphin is able to pick up the data before it expires, i.e., if no dolphin moves within the communication range of the generating sensor between the moment the event is sensed and its expiration time, then the data is lost. We briefly remark that this harvesting process is stochastic in nature. In fact, dolphins have no information about event generation and expiration periods. As a consequence, they are not able neither to establish an effective moving path, nor to correct their trajectory according to the generated events. Therefore, they perform random walks through the field and opportunistically move within sensor neighborhoods. If visited sensors have collected undelivered and not yet expired events, these are sent to the dolphin. Since event generation and expiration are random processes, this configures as a stochastic game.

The final goal of this analysis is to determine how the data gathering probability increases by increasing the number of dolphins k, given the parameters λ and *buf size* or μ.

Geometrically, we could formulate the problem in a 2-dimensional space as follows. The dolphins are initially uniformly deployed in the field A; then, as they move, they cover the area they span. This amounts to the whole area covered by the radio range of the dolphins as they move. If all events were generated initially and never expired, then the problem of determining the gathering probability would be easily solved: in that case only the gathering time would be dependent on the number of dolphins k. If instead the data can be freshly generated, then some sites need to be visited periodically[2], so that nodes can deliver their fresh data to dolphins. We take this phenomenon into account by reverting the area in the radio range of the sensor to the uncovered state. Finally, the event expiration rate imposes additional constraints on the visiting time. In our analogy, that corresponds to a temporary reversion of the area to the uncovered state. If no dolphin reaches the sensor area before the event expires, then the area turns back to covered but the event is definitely loss.

Through all this section, for easy of calculation, we consider a one dimensional approximation of this two-dimensional problem. Let us observe that, even if this approximation seems oversimplifying, its ability to catch the essential principles has been advocated in [7].

B. One-dimensional Analysis

In one-dimensional analysis, since every dolphin has a fixed coverage range equal to r, we can consider the dolphins as r-long segments that aim at covering a line of size L, where L is

[2]Note that the period is not fixed and is mainly dependent on the event generation rate.

the linear dimension of the square *A*. Thus, the determination of the probability that *k* dolphins collect a certain fraction of generated events within a time interval *T* can be traced to the portion of line *L* covered by *k* moving segments in *T* seconds. Our research plans will investigate the analytic model by finally resolving a sequence of problems of increasing difficulty. In particular, we will follow three main directions of simplification:

1) In this paper we analyze a single-dolphin case. We argue this provides useful insights on the multiple-dolphins problem. Our future research work will extend our investigations to the case of *k* dolphins.
2) With regards to mobility, first we consider a simplified model where nodes can jump along the line, without any constraints. Later, to obtain more complete and realistic results, we introduce random walks according to the model presented in Section III. In this model, each segment moves *T*v* space left or right at any time interval *T*.
3) In this paper we simplisticly assume that data are generated at time 0 by all nodes. Our future resarch work will explore how the generation rate λ and the expiration interval (including the parameters *buf size* or μ) impact the results.

Let us observe that, for the sake of simplicity, we will perform a discrete-time analysis. The following section investigates the time needed for a single dolphin performing a random walk to cover the whole line.

B.1. Single-dolphin analysis

First, let us suppose that a single dolphin with coverage range[3] *r* moves within the line *l* by performing *random jumps*, i.e., leaping from a position to the subsequent without any constraints. In this analysis we aim to determine what is the expected number of trials needed for the dolphin to cover the whole line *l*. This configures as a well-known "coupon collector problem". A simple solution can be found by discretizing the line into $N/2r$ portions, and investigating the average time needed for a leaping agent to visit all the portions. Poisson process theory [19] proves that

$$E[T] \;=\; \int_0^\infty \left\{ 1 - \prod_{j=1}^{m} \left(1 - e^{-p_j t}\right) \right\} dt \qquad (1)$$

where p_j denotes the probability of collecting one coupon. In our case $p_j = 2r/L$.

As a further step we introduce an additional constraint, i.e., the dolphin performs only random walks. This way, the dolphin is no more allowed to leap from a position on the line *l* to another randomly chosen at each timestep. By performing a random walk the dolphin is constrained to move one step in a random direction at each timestep. This means that if the dolphin is at position x at time T_1, at $T_1 + T$ it is allowed

[3]This will be represented as a segment of length *2r* centered at the current position of the dolphin

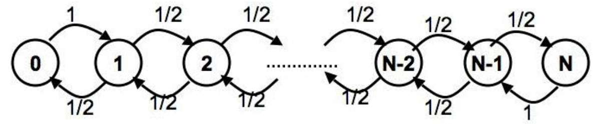

Fig. 7. Markov chain representing dolphin random walk

to move only toward positions $x - 1$ or $x + 1$. Under these conditions we aim to solve the same problems as above.

As a traditional stochastic process, random walks have been deeply investigated with the help of Markov chain models. We argue Markov chains are suitable to solve our problem. Let us consider the line *l* discretized in a number *N* big at will of small segments of identical size *m*. Then, the dolphin performs a random walk by moving one segment left or right from its present position at each timestep.[4] Obviously, the length of a timestep *t* is related to the speed *v* of the dolphin and to the size of the segments *m*: $t = \frac{m}{v}$. As a matter of facts, the dolphin chooses the next position on the line (i.e., next state) by considering only its present position (i.e., current state). The history of its movements has no impact on its choice. This suggests us to match each segment on *l* with a different state on a *N+1*-state Markov chain (as in Figure 7). At every timestep the dolphin moves toward one of the two states adjacent to its current with identical probability. The motion is only bounded at the endpoints of *l*.

We aim to determine the number of steps the dolphin needs to cover the whole line: if we consider a dolphin starting from a random position on *l*, that is essentially the time the dolphin needs to visit the second endpoint (whichever is the visiting order of the endpoints). Then, we can obtain the final result by averaging the expected time to cover the whole line on the *N* different starting points (by considering for any state the probability for it to be a starting point, which we assume to be uniform, i.e., $\frac{1}{N}$).

As a first step, let us assume that the dolphin is in state 0 and calculate how many steps will it take to move to state N (i.e., to reach the other endpoint of the line). By considering Figure 7, it is easy to prove that, for any state i, this time is equal to:

$$\begin{cases} \mu(0,N) \;=\; 1 \;+\; \mu(1,N) \\ \mu(i,N) \;=\; 1 \;+\; \frac{\mu(i-1,N)}{2} \;+\; \frac{\mu(i+1,N)}{2} \\ \qquad\qquad\qquad \forall i : 1 \le i \le N-2 \\ \mu(N-1,N) \;=\; 1 \;+\; \frac{\mu(N-2,N)}{2} \end{cases} \qquad (2)$$

where $\mu(i, N)$ represents the average time the dolphin needs to move from state i to state N. Intuitively, from any inner state i, the dolphin will need 1 step, plus the sum of the average time needed from any of its adjacent states times the probability

[4]We rapidly observe that the length of a single segment has no relationship with the communication range of the dolphin; in particular the former can be much smaller than the latter. The dolphin is allowed to cover many segments simultaneously, while it is allowed to move only one segment per timestep.

of reaching that state (in our case it is considered equal to $\frac{1}{2}$). Thus, the solution to the problem is the solution of this N-equations system in N unknowns ($\mu(0,N)...\mu(N-1,N)$). We can easily find it by re-writing the system in matricial form $A \cdot x = \mathbf{1}$, and then find $x = A^{-1} \cdot \mathbf{1}$, where

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots & & & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & & & 0 \\ 0 & & & \cdots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & & & \cdots & & 0 & -\frac{1}{2} & 1 \end{pmatrix}$$

is the $N \times N$ matrix, x is a $N \times 1$ array, and $\mathbf{1}$ represents an $N \times 1$ array with all elements equal to 1.

$x = A^{-1} \cdot \mathbf{1}$ means that any element of the solution x is the sum of all the elements in the corresponding row of A^{-1}. We argue that A^{-1} has a regular form (some interesting studies on the inversion of tridiagonal matrices can be found in [10]). This permits us to draw the following solution:

$$x_i \;=\; N \;-\; i \;+ 2 \cdot \sum_{k=i}^{N-i-1} k \;+ i \cdot (N-i) \cdot 2 \qquad \forall i : 0 \leq i \leq N-1 \tag{3}$$

For $i = 0$ this equation gives us the average number of timesteps needed to move from state 0 to state N, which is equal to N^2.

Now we need to find the time needed to reach the first endpoint (states 0, N). To prove this result, we can adopt a proof strategy similar to the one just shown. This problem can be easily mapped in the Markov chain in Figure 8. In fact, since we aim to determine the time needed to reach the first endpoint 0 or N, then these states are absorbing states.

By considering Figure 8 it is easy to prove that, for any state i, this time is equal to

$$\begin{cases} \mu(0,N) \;=\; \mu(N,N) \;=\; 0 \\ \mu(i,N) \;=\; 1 \;+\; \frac{\mu(i-1,N)}{2} \;+\; \frac{\mu(i+1,N)}{2} \\ \qquad \forall i : 1 \leq i \leq N-1 \end{cases} \tag{4}$$

This is a N-1-equation system in N-1 unknowns. Analogously to the previous case, it is easy to find the solution by rewriting the system in matricial form $B \cdot y = \mathbf{1}$, and then find $y = B^{-1} \cdot \mathbf{1}$, where

$$B = \begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 & \cdots & & & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & & & 0 \\ 0 & & & \cdots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & & & \cdots & & 0 & -\frac{1}{2} & 1 \end{pmatrix}$$

is the $N \times N$ matrix, y is a $N \times 1$ array, and $\mathbf{1}$ represents an $N \times 1$ array with all elements equal to 1. $y = B^{-1} \cdot \mathbf{1}$ means that any element of the solution y is the sum of all the elements in the corresponding row of B^{-1}. Also in this case B^{-1} has a regular form. This permits us to draw the following solution:

$$Z_i \;=\; i \cdot (N-i) \;\; \forall i : 0 \leq i \leq N \tag{5}$$

As a consequence the average time needed to visit all the states of the chain, by starting from state i is

$$T_i \;=\; Z_i \;+\; N^2 \;\; \forall i : 0 \leq i \leq N \tag{6}$$

Averaged on the number of states this leads to the following result:

$$E[T] \;=\; N \;+\; N^2 \tag{7}$$

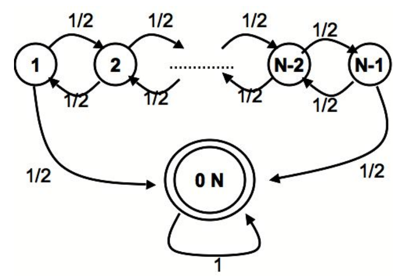

Fig. 8. Markov chain representing the dolphin path toward the first visited enpoint

B.2. Evaluation of the Analysis

With the goal of obtaining a quick coarse-grained evaluation of our analytic results, we ran several Matlab simulations. In particular, we investigated the correctness of the results about dolphin performing random walk. We performed simulations by measuring the time needed by a node to cover all the states of a chain, i.e., to pass through all the positions performing random walks, by starting from a different position. We considered a 20-state chain and we repeated 1000 simulation trials for each starting position of the dolphin. In Figure 9 x-axis denotes the starting position i, i.e., the starting state of the Markov chain, y-axis instead denotes the number of steps needed to cover the whole line. Experimental results match very well formula (6).

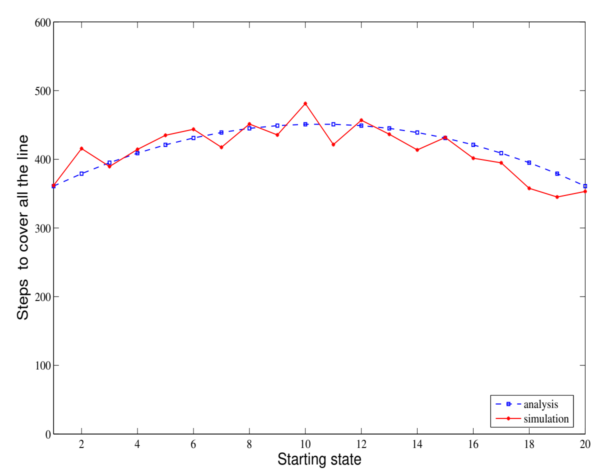

Fig. 9. Time needed for a dolphin to cover the whole line

VI. Conclusions

In this paper we proposed DDD, an approach to apply delay-tolerant networking in the underwater environment with extremely constrained communication resource, requiring energy-expensive transmission devices. In DDD, a few capable agents scour the field and collect information sensed by a number of distributed resource-limited devices. These sensors upload sensed events only when a dolphin is in direct one-hop range, to minimize the energy required for communications. Experimental results investigate how system parameters (e.g., the number of dolphins) should be tuned to obtain the desired average delay and ratio of harvested events, thus demonstrating the feasibility of the approach with very limited numbers of dolphins in most application scenarios. Our on-going research work mainly follows two directions. First, we are investigating more accurately the energy balance of different communication solutions (different modem choices, MAC layer protocols, 1-hop vs. multi-hop relaying). Second, we are continuing to work on the analytical model of the system, to obtain an expression of its main performance indicators.

REFERENCES

[1] I. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. In *Elsevier Journal of Ad Hoc Networks*, volume 3, pages 257–281, Mar. 2005.

[2] S. Burleigh, K. Fall, V. Cerf, R. Durst, K. Scott, H. Weiss, L. Torgerson, and A. Hooke. Delay-Tolerant Networking: an Approach to Interplanetary Internet. In *IEEE Communications Magazine*, volume 41, pages 128–137, Jun. 2003.

[3] T. Clausen, P. Jacquet, and L. Viennot. Comparative study of cbr and tcp performance of manet routing protocols. In *IEEE MESA*, Jan. 2002. ETSI.

[4] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications. In *IEEE Network*, volume 20, pages 12–18, May-Jun. 2006.

[5] K. Fall. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM'03*, Aug. 2003.

[6] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The whoi micro-modem: An acoustic communications and navigation system for multiple platforms. In *IEEE/MTS OCEANS Conference*, Sept. 2005.

[7] A. Goel, S. Rai, and B. Krishnamachari. Sharp thresholds For monotone properties in random geometric graph. In *ACM STOC04*, Jun. 2004.

[8] F. Harary. *Graph Theory*. Addison Wesley Publishing, 1995.

[9] J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Underwater sensor networking: Research challenges and potential applications . In *USC/ISI Technical Report ISI-TR-2005*, 2005.

[10] Y. Huang and W. McColl. Analytical Inversion of General Tridiagonal Matrices. In *Journal of Physics A: Mathematical and Generals*, volume 30, pages 7919–7933, Nov. 1997.

[11] R. Humston, D. B. Olson, and J. S. Ault. Behavioral Assumptions in Models of Fish Movement and Their Influence on Population Dynamics. In *Transactions of the America Fisheries Society*, volume 10, pages 1304–1328, Jun. 2004.

[12] D. B. Kilfoyle and A. B. Baggeroer. The State of the Art in Underwater Acoustic Telemetry. *IEEE Journal of Oceanic Engineering*, OE-25(1):4–27, January 2000.

[13] J.-Y. LeBoudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. In *IEEE Infocom '05*, Mar. 2005.

[14] LinkQuest, Inc. Underwater Acoustic Modems. `http://www.link-quest.com/html/uwm_hr.pdf`.

[15] Crossbow Mica 2 Motes. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_OEM_Edition_Datasheet.pdf`.

[16] S. PalChaudhuri, J.-Y. LeBoudec, and M. Vojnovic. Perfect Simulations of Random Trip Models. In *38th Annual Simulation Symposium*, Apr. 2005.

[17] J. Proakis, E. Sozer, J. Rice, and M. Stojanovic. Shallow Water Acoustic Networks. In *IEEE Communications Magazine*, volume 39, pages 114–119, Nov. 2001.

[18] The Qualnet Simulator. `http:://www.scalable-networks.com`.

[19] S. Ross. *Introduction to Probability Models*. Academic Press (Elsevier), 2003.

[20] T. Small and Z. J. Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where There is a Whale, There is a Way). In *ACM MOBIHOC'03*, Jun. 2003.

[21] L. F. M. Vieira, J. Kong, U. Lee, and M. Gerla. Analysis of Aloha Protocols for Underwater Acoustic Sensor Networks. In *ACM WUWNET'06 - Poster available at* `http://wuwnet.engr.uconn.edu/wipposters/luiz_aloha.pdf`, Sept. 2006.

[22] J. Wills, W. Ye, and J. Heidemann. Low-Power Acoustic Modem for Dense Underwater Sensor Networks. In *ACM WUWNET'06*, Sept. 2006.

[23] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *ACM MOBIHOC'04*, May 2004.