Path Finding for Maximum Value of Information in Multi-modal Underwater Wireless Sensor Networks

Petrika Gjanci, Chiara Petrioli, Stefano Basagni, Cynthia A. Phillips, Ladislau Bölöni and Damla Turgut

Abstract—We consider underwater multi-modal wireless sensor networks (UWSNs) suitable for applications on submarine surveillance and monitoring, where nodes offload data to a mobile autonomous underwater vehicle (AUV) via optical technology, and coordinate using acoustic communication. Sensed data are associated with a value, decaying in time. In this scenario, we address the problem of finding the path of the AUV so that the *Value of Information* (VoI) of the data delivered to a sink on the surface is maximized. We define a *Greedy and Adaptive AUV Path-finding* (GAAP) heuristic that drives the AUV to collect data from nodes depending on the VoI of their data. For benchmarking the performance of AUV path-finding heuristics, we define an integer linear programming (ILP) formulation that accurately models the considered scenario, deriving a path that drives the AUV to collect and deliver data with the maximum VoI. In our experiments GAAP consistently delivers more than 80% of the theoretical maximum VoI determined by the ILP model. We also compare the performance of GAAP with that of other strategies for driving the AUV among sensing nodes, namely, random paths, TSP-based paths and a "lawn mower"-like strategy. Our results show that GAAP always outperforms every other heuristic in terms of delivered VoI, also obtaining higher energy efficiency.

Index Terms—Underwater networking, Value of Information, Autonomous Underwater Vehicle, multi-modal communications

1 Introduction

The ever-alive quest about its own origins and strengths has brought humanity to explore Earth's most remote lands and even the farthest planets. Yet, we know very little about our oceans and what lies beneath them. This is a particularly relevant knowledge gap, as underwater exploration and monitoring has emerged as a vital part of the economy and of the safety infrastructure of many countries. Aquaculture, coastal surveillance and protection, monitoring of oil industry deployments, telecommunications, pollution and climate control, search missions and preservation of cultural heritage are just a few of the many applications that will be enabled by the exploration and sustainable exploitation of the underwater world [1]. Currently, underwater systems rely on tethered vehicles, cabled monitoring stations, or leaving instrumentation on site and then retrieving it periodically: All costly propositions. Only recently, advances in wireless acoustic and optical underwater communications have made underwater monitoring applications feasible and cost effective. Organized in an *Underwater Wireless Sensor Network* (UWSN), nodes can be deployed to cover large areas and can route data efficiently directly or through mobile *autonomous underwater vehicles* (AUVs) to a data collection center. The most difficult obstacles to the realization of these networks concern the fact that the two viable alternatives for underwater communications, namely, acoustic and optical, impose contrasting challenges. Acoustic communications provide long-distance connectivity (e.g., kilometers) but are bandwidth-limited (currently, few hundreds of bit per second). This is often insufficient for emerging applications that produce large amounts of data, including high definition pictures and video transmissions. Optical communication obtains high bandwidth connectivity (up to 10Mbps) with energy-per-bit orders of magnitude lower than that of acoustic communications [2]. However, it allows nodes to robustly communicate only when they are few meters from each other (usually less than 10m) [3], [4]. Researchers are recently agreeing that enhancing the performance of UWSNs and enabling critical applications requires multi-modal communication, i.e., a judicious combination of acoustic and optical information exchange [5], [6], [7], [8]. If an AUV is available for network operations, data exchange can be performed effectively through wireless optical technology, as the AUV can hover close enough to the node to make optical transfer possible at high data rate [8], [9], [10]. In such a scenario, the AUV could be made aware of a node sensing relevant events through acoustic communication, so that it can travel to that node to collect sensed data and then surface to transfer them to the network data collection center. This operation should be done swiftly, so that it does not affect the value of the data as perceived by the user, in that such value typically decays in time [11]. Typical examples include real-time video from monitoring valuable assets, such as cultural

- *P. Gjanci and C. Petrioli are with the Department of Computer of the Università di Roma "La Sapienza," Roma, Italy.*
  E-mail: {gjanci,petrioli}@di.uniroma1.it
- *S. Basagni is with the ECE Department of Northeastern University, Boston, MA.*
  E-mail: basagni@ece.neu.edu
- *C. A. Phillips is with the Center for Computing Research of Sandia National Laboratories, Albuquerque, NM.*
  E-mail: caphill@sandia.gov
- *L. Bölöni and D. Turgut are with the EECS Department, University of Central Florida, Orlando, FL.*
  E-mail: {turgut,lboloni}@eecs.ucf.edu

This work was supported in part by the EU FP 7 ICT project SUNRISE "Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet." Stefano Basagni was partially supported by the NSF grant CNS 1428567 "MRI: Development of the Northeastern University Marine Observatory NETwork (NU MONET)," and by a GENI SAVI travel grant. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-NA0003525. SAND2017-4391 C.

heritage relics, CO_2-filled boreholes, oil wells, etc. The value of a chunk of data can be defined intuitively through the benefits obtained by taking an action based on it. For instance, if an oil spill is reported, the owner of the oil rig can timely intervene and save the costs of further damage and environmental cleanup. Thus, the value of this notification is proportional to the money saved through the timely intervention. Naturally, the earlier one can intervene, the better. This is why, in most situations, the VoI is largest at the very moment an event is sensed and then it decreases in time. Different events will have different initial VoI, and their value may decrease differently as a function of their urgency. For example, information on oil spills is urgent and its value decreases over time spans of few tens of minutes. On the other hand, the value of information about signs of corrosion of underwater pipes decreases at a slower pace.

This paper proposes a new approach to the problem of delivering data with high *Value of Information* (VoI) to the user by designing a distributed AUV routing heuristics to enable UWSN applications. Fig. 1 illustrates the type of UWSNs we consider in this paper. Sensor nodes deployed underwater monitor assets or conditions. They sense and record data, waiting for an AUV to arrive to collect them optically. Nodes are also equipped with acoustic modems to exchange control information with the AUV. For instance, if a node has important data to deliver, it can send a short control packet over the acoustic channel to the AUV. Periodically, the AUV resurfaces and wirelessly (RF) transmits the collected data to a collection point (a *sink*), located on the surface.

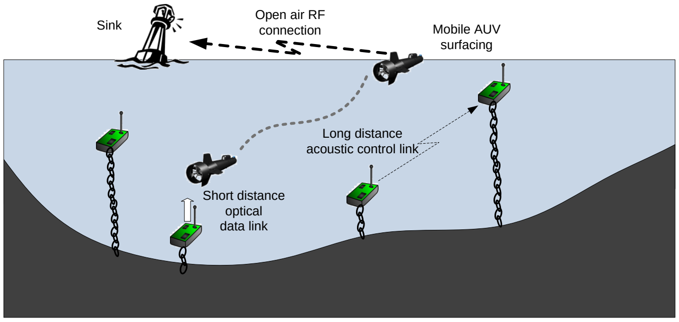

Sink
Open air RF connection
Mobile AUV surfacing
Long distance acoustic control link
Short distance optical data link

Fig. 1: A UWSN where nodes are visited by an AUV.

Data produced by a node sensing an event vary in size, value and urgency, according to application requirements. The VoI of the data from an event is highest at the moment the event is detected, and may decay with time. Therefore, data reporting an event should be delivered to the sink as soon as possible: The later the data reaches the sink, the lower their value, if any. Our goal is to investigate the theoretical and practical challenges of finding the path of the AUV that maximizes the total VoI delivered to the sink. The contributions of this paper are the following:

1) We define a new Integer Linear Programming (ILP) model for finding AUV paths that maximize the VoI of data delivered to the sink. Our model provides provable bounds on the best possible network performance (e.g., the best achievable VoI) for benchmarking distributed protocols. The model is independent of sensor deployment strategies, and has parameters for controlling data generation rate, data transmission rates, and AUV speeds. Our solution allows us to compute an upper bound on the maximum VoI retrievable from networks whose size is comparable to that of actual (4 to 9 nodes) and desirable (12 to 35 nodes) UWSNs. To the best of our knowledge this is the first model for finding AUV path that takes into account the VoI.
2) We define a realistically deployable heuristic for AUV path finding that adapts to events occurring at unpredictable locations and times. The AUV chooses the next node to be visited based on the VoI it expects to collect at the next location. The information needed to make this decision is propagated to the AUV using short *event packets* transmitted acoustically. The AUV plans to visit a node that has sent an event packet if and only if visiting that node increases the VoI of the data it will deliver to the sink. Because it makes decisions based on what is best at the moment, and adapts the path finding process to new information, we call our heuristic *Greedy and Adaptive AUV Path* finding, or GAAP for short.
3) GAAP effectiveness in delivering data with high VoI is demonstrated by evaluating its performance in UWSN scenarios where we vary the number of nodes, events occurrence (i.e., traffic) and event value. We start by comparing the performance of GAAP with that of the ILP-based upper bound, termed OPT. Our experiments show that GAAP achieves VoI that is remarkably close to that achieved by OPT. In fact, in scenarios with variable number of nodes (from 4 to 35) and events, we observe that GAAP always delivers more than 80% of the theoretical maximum VoI obtained by OPT. We also demonstrate the benefit of VoI awareness through a simulation-based comparative performance evaluation of GAAP against three non-VoI-aware heuristics where the AUV visits nodes following random paths, TSP-based paths or travels to the nodes according to a "lawn mower"-like strategy. Our results show clearly that, by explicitly considering VoI for finding paths, GAAP drives the AUV along paths that are very similar to those determined by the all-knowing OPT. Moreover, we observe that GAAP always outperforms every other heuristic. Specifically, we found that irrespective of network size and event occurrences GAAP delivers from 57% to 77% more VoI than that delivered by the other three solutions. It is also much more energy effective than the non-VoI-aware solutions in that it requires up to 70% less energy to deliver the same amount of VoI.

A preliminary version of this work was presented in a prior conference paper [12]. The present paper extends the work of that prior research greatly. Differences include a thorough revision of the ILP formulation that now considers fine grain timing and achieve remarkably higher scalability: An optimal solution can now be found for networks of up to 35 nodes instead of just 12. GAAP has been extensively changed to consider different types of events (with varying VoI decays) and also changes made on event notification and data chunk generation time. The performance evaluation section has been re-organized, and includes comparisons among GAAP and multiple heuristics, now made event-aware for greater fairness and further insights. Results on energy consumption/efficiency presented here were not present in our previous paper.

The rest of the paper is organized as follows. Section 2 defines the problem we consider in detail. In Section 3 we present an ILP formulation for AUV path finding which scales to networks with tens of nodes. Our heuristic for AUV routing (GAAP) is presented in Section 4. Section 5 presents the performance evaluation and comparisons between GAAP, OPT and three different non-VoI-aware solutions. Section 6 reviews literature on the topics of this paper. Finally, Section 7 concludes the paper.

2 Problem Definition

The scenarios we consider comprise a set S of (sensor) nodes $S_1, \ldots, S_{|S|}$ statically deployed underwater in 3D. Nodes perform surveillance operations for a given time T. The location of each node is known (e.g., from manual deployment or by using localization techniques). The nodes perform continuous sensing (e.g., taking videos), with node S_i storing the sensed data chunk p_{it} at time t, $0 \leq t < T$. The data $p_{i\tau}$ observed by a node S_i at a given time τ has a *value of information* $\mathcal{V}_{p_{i\tau}}(t)$, at time $t \geq \tau$. The function $\mathcal{V}_{p_{i\tau}}(t)$ is non-increasing in t. The VoI of a data chunk is highest at the moment when it is sensed, when its value is $\mathcal{V}_{p_{i\tau}}(\tau)$; this base value varies depending on the importance of the information captured in the data chunk.

Throughout the time of network operations an Autonomous Underwater Vehicle (AUV) is deployed to visit the nodes to collect the sensed information via high-data-rate optical communication. The AUV periodically surfaces to offload what it has collected to a data collection point (a *sink*). Communications between the sink and the AUV happen at high data rate, wirelessly. To ensure wireless connectivity with the sink, the AUV must choose among a set of possible surfacing locations W. The path the AUV follows is a sequence of *runs* in each of which the AUV visits a number of nodes to collect some of their data chunks and surfaces to report them to the sink. Specifically, during the kth run, the AUV makes a set of visits $\{\ldots(S_i, t_i^k)\ldots\}$, collecting the last data chunk from node S_i at time t_i^k. Finding the path that yields the maximum VoI is done by first defining the VoI of the data chunks collected from a node S_i when the AUV travels a given path P, and then choosing the path P that maximizes the VoI of data chunks collected from all nodes. It is done as follows.

Let $R = \{1, \ldots, r\}$ be the set of runs performed by the AUV by time T. Node S_i is visited by the AUV during a set of runs $R_i \subseteq R$. (The AUV might not visit every node in every run, so in general $|R_i| \leq r$.) Consider a run $k \in R_i$. Let $\text{pred}(k, R_i)$ be the last run the AUV visited node S_i before run k. Thus, if the runs in R_i are sorted by time, $\text{pred}(k, R_i)$ is the largest run in R_i less than k. (By definition $t_i^{pred(k,R_i)} = 0$, with k being the smallest element of R_i.) The value of information of the data sensed by node S_i and delivered to the sink by the AUV traveling path P during time T is given by summing the values of information collected by S_i between two consecutive visits of the AUV when each packet is delivered to the sink. More precisely:

$$\mathcal{V}(S_i, P) = \sum_{k \in R_i} \sum_{h = t_i^{pred(k,R_i)}}^{t_i^k - 1} \mathcal{V}_{p_{i,h}}(t_{i,h}), \tag{1}$$

where $t_{i,h}$ is the time the AUV delivers the packet collected from node S_i at time h to the sink. The AUV must be on the surface to deliver packets to the sink. After each run, it delivers the packets it has collected during the run.

The AUV path finding problem is then stated as follow: *Given $|S|$ nodes and their locations, given a set of surfacing locations, and given the value of information of the sensed data, determine the path P_{opt} (sequence of nodes and surfacing locations) of the AUV so that the value of information is maximized:*

$$P_{opt} = \underset{P}{\operatorname{argmax}} \left(\sum_{i=1}^{|S|} \mathcal{V}(S_i, P) \right). \tag{2}$$

We assume the AUV begins and ends on the surface, since any collected but undelivered packets would be worth nothing.

3 A Mathematical Model for AUV Path Finding

We model the problem defined in Section 2 by the following Integer Linear Programming formulation.

Parameters

- T is the length of network operations, divided into time units numbered from 0 to $T-1$. When a node optically transmits a data chunk to the AUV at time $t < T-1$ the AUV has the data chunk at time $t+1$. Similarly for wireless transmissions from the AUV to the sink. The last time the sink can receive data is T (for data transmitted by the AUV at time $T-1$).
- S is the set of nodes (and their locations). We use the letters i and j to indicate generic nodes (their location).
- W is the set of surfacing locations.
- $N = S \cup W$ is the set of all locations to which the AUV can travel and sojourn to either receive or transmit data chunks, respectively.
- ω is a fictitious location, indicating in fact that the AUV is in transit, i.e., it is not at any actual location in N.
- $\delta_{i,j}$ is the time it takes for the AUV to travel between any two locations i and j in N, in time units. The time distance $\delta_{i,j}$ is easily derived from the known position of nodes and surfacing locations and from the known speed of the AUV. When the AUV leaves location i at time t, it arrives at location j at time $t + \delta_{i,j}$ and can take action at node j during time $t + \delta_{i,j}$.
- τ_i is the shortest travel time for the AUV to go from node $i \in S$ to some location $w \in W$ on the surface and vice versa (assumed symmetrical), in time units.
- $d_{i,p}$ is the pth *data chunk* (packet) produced at node $i \in S$.
- $b_{i,p}$ is the release (beginning) time of data chunk $d_{i,p}$, when its recording finishes and it is made available.

- $e_{i,p}$ is the deadline (end) for data chunk $d_{i,p}$ after which the value of information carried by that data chunk becomes 0. Deadline $e_{i,p}$ will generally depend on node i and capture time $b_{i,p}$. It can also depend upon what is happening (such as a first detection of important motion) and/or other properties of the data/image. Times $b_{i,p}$ and $e_{i,p}$ are the beginning and ending times respectively of the interval where the pth data chunk from node i is relevant for scheduling the AUV.
- *TimeToTransfer()* is a function that given a data chunk returns the time needed to transfer it to the AUV. This time depends on the optical communication rate and on the size of the data chunk.[1]
- *TimeToBroadcast()* is a function that given a data chunk returns the time needed to broadcast it to the sink. This time depends on the wireless communication rate and on the size of the data chunk.
- $\mathcal{V}^t_{i,p}$ is the value of information of the data chunk $d_{i,p}$ delivered at time t.
- $G = \{d_{i,p} \mid b_{i,p} \in [1 \dots T - \tau_i - 2]\}$ is the set of all data chunks that can be delivered to the sink by time T.
- $D(d_{i,p})$ is the set of all legal delivery times for data chunk $d_{i,p} \in G$. After collection, a data chunk should be delivered before it expires and by time T. In other words, $D(d_{i,p}) = \{t \mid t \in [b_{i,p} + \tau_i + 2, T] \wedge t \leq e_{i,p}\}$.
- $C(d_{i,p})$ is the set of legal collection times for $d_{i,p} \in G$. A data chunk can be collected by the AUV if there is enough time to deliver it to the sink before its expiration deadline, and by time T. Formally, $C(d_{i,p}) = \{t \mid t \in [b_{i,p} + 1, T - \tau_i - 1] \wedge t \leq e_{i,p} - \tau_i - 1\}$.
- F and K are sets that combine all data chunks and their legal delivery and collection times, respectively: $F = \{(d_{i,p}, t) \mid d_{i,p} \in G \wedge t \in D(d_{i,p})\}$ and $K = \{(d_{i,p}, t) \mid d_{i,p} \in G \wedge t \in C(d_{i,p})\}$. We use $F(t)$ to denote all data chunks that can be delivered to the sink at time t, and $K(i,t)$ to denote all data chunks that can be collected from node i at time t.
- $L(n)$ is the set of times where it would make sense for the AUV to be at location $n \in N \cup \{\omega\}$. When $n \in S$, we have $L(n) = \{t \mid t \in [\tau_n, T - \tau_n - 1]\}$; for $n \in W$, we have $L(n) = \{t \mid t \in [0, T]\}$, and $L(\omega) = \{t \mid t \in [1, T - 2]\}$.
- $\Gamma = \{(i,j,t) \mid \forall i,j \in N \wedge \neg(i \in W \wedge j \in W),\ t \in L(i) \wedge t + \delta_{i,j} \in L(j)\}$ is the set of all legal triples (i,j,t) modeling the movements in time of the AUV. The triple (i,j,t) indicates that at time t the AUV departs node i for node j (if it is allowed to be at node i at time t and at node j at time $t + \delta_{i,j}$). As the AUV transmits all collected data after surfacing, it does not move to other surface locations directly without first visiting some underwater nodes.

Variables

- $x^t_{i,p}$: Binary variable taking the value 1 if data chunk $d_{i,p}$ is successfully delivered to the sink at time $t \leq e_{i,p}$; 0 otherwise.
- $c^t_{i,p}$: Binary variable taking the value 1 if a data chunk $d_{i,p}$ is collected at time t; 0 otherwise.
- $\ell_{i,j,t}$: Binary variable taking the value 1 if the AUV departs from (leaves) location i at time t directed towards j; 0 otherwise.
- $a_{i,j,t}$: Binary variable taking the value 1 if the AUV arrives at node j at time t coming from node i; 0 otherwise.
- $z_{n,t}$: Binary variable taking the value 1 if the AUV is at location n at time t, $n \in N \cup \{\omega\}$; 0 otherwise.
- y_t: Binary variable taking the value 1 if the AUV is on the surface (at one of the locations in W) at time t; 0 otherwise. Helper variables y_t make the model description more succinct. They are completely determined by the z variables: $y_t = \sum_{w \in W} z_{w,t}$.

ILP formulation

The objective function maximizes the value of information collected from all nodes and delivered by time T. Variable $x^t_{i,p}$ only contributes to the objective function when it is equal to 1, that is when data chunk $d_{i,p}$ is delivered at time t. It contributes the value associated with that delivery.

$$\text{maximize} \sum_{(d_{i,p},t) \in F} \mathcal{V}^t_{i,p} \cdot x^t_{i,p}$$

subject to the following constraints.

$$\sum_{n \in N \cup \{\omega\}} z_{n,t} = 1 \quad \forall t \in [1, T-1] \tag{3}$$

$$\sum_{w \in W} z_{w,0} = 1 \tag{4}$$

$$\sum_{w \in W} z_{w,T} = 1 \tag{5}$$

$$\ell_{i,j,t} = a_{i,j,t+\delta_{i,j}} \quad \forall (i,j,t) \in \Gamma \tag{6}$$

$$\begin{aligned} \sum_{j \in N \setminus \{i\} \wedge (j,i,t-\delta_{i,j}) \in \Gamma} a_{j,i,t} \ + \sum_{j \in N \setminus \{i\} \wedge (i,j,t) \in \Gamma} \ell_{i,j,t} \leq z_{i,t} \\ \forall i \in N,\ t \in L(i) \end{aligned} \tag{7}$$

$$\begin{aligned} z_{i,t} = z_{i,t-1} + \sum_{j \in N \setminus \{i\} \wedge (j,i,t-\delta_{i,j}) \in \Gamma} a_{j,i,t} \ - \\ \sum_{j \in N \setminus \{i\} \wedge (j,i,t-1) \in \Gamma} \ell_{i,j,t-1} \\ \forall i \in N,\ \forall t \mid t-1 \in L(i) \wedge t \in L(i) \end{aligned} \tag{8}$$

$$z_{i,\tau_i} = \sum_{j \in W \ \wedge \ \delta_{i,j} = \tau_i} a_{j,i,\tau_i} \quad \forall i \in S \tag{9}$$

$$x^t_{i,p} \leq y_t \quad \forall (d_{i,p}, t) \in F \tag{10}$$

$$c^t_{i,p} \leq z_{i,t} \quad \forall (d_{i,p}, t) \in K \tag{11}$$

$$x^t_{i,p} \leq \sum_{\tau = r+1}^{t - \tau_i - 1} c^\tau_{i,p} \quad \forall (d_{i,p}, t) \in F \tag{12}$$

$$\sum_{t \in D(d_{i,p})} x^t_{i,p} \leq 1 \quad \forall d_{i,p} \in G \tag{13}$$

[1] We assume that a data chunk can be transferred through optical and wireless communication well within one unit of time. In this case, *TimeToTransfer()* and *TimeToBroadcast()* will return the fraction of time unit needed to transfer the data chunk they are applied to.

$$\sum_{d_{i,p} \in F(t)} \mathit{TimeToBroadcast}(d_{i,p}) \cdot x^t_{i,p} \leq y_{t-1}$$

$$\forall t \; s.t. \; \exists (d_{i,p}, t) \in F \tag{14}$$

$$\sum_{d_{i,p} \in K(i,t)} \mathit{TimeToTransfer}(d_{i,p}) \cdot c^t_{i,p} \leq z_{i,t-1}$$

$$\forall i \in S, \forall t \; s.t. \; \exists (d_{i,p}, t) \in K \tag{15}$$

$$x^t_{i,p} \in \{0,1\} \quad \forall (d_{i,p}, t) \in F \tag{16}$$

$$c^t_{i,p} \in \{0,1\} \quad \forall (d_{i,p}, t) \in K \tag{17}$$

$$z_{n,t} \in \{0,1\} \quad \forall n \in N \cup \{\omega\}, t \in L(n) \tag{18}$$

$$\ell_{i,j,t} \in \{0,1\} \quad \forall (i,j,t) \in \Gamma \tag{19}$$

$$a_{i,j,t} \in \{0,1\} \quad \forall \exists (i,j,t-\delta_{i,j}) \in \Gamma \tag{20}$$

$$y_t = \sum_{w \in W} z_{w,t} \quad \forall t \in [0, T-1] \tag{21}$$

The first three sets of constraints restrict the $z_{n,t}$ variables that indicate AUV locations. Constraints (3) enforce the physical requirement that in any real path, an AUV is in exactly one place (or in transit) at any given time. That is, Constraints (3) forbids $z_{n_1,t} = z_{n_2,t} = 1$ for $n_1 \neq n_2$. Constraint (4) is the special case of Constraints (3) for time 0. The AUV must start on the surface, that is, be at a node in W at time 0. Since all travel times $\tau_i \geq 1$, we have $0 \notin L(n)$ for any $n \notin W$. Thus the range of locations in the sum can be restricted to elements of W. Constraint (5) is the special case of Constraints (3) at the end, time T, where the AUV must also be at the surface.

The following four constraints ensure that travel times are consistent with the physical reality of a single AUV visiting locations. Constraints (6) ensure that if the AUV leaves node i at time t for destination node j it takes a most direct route and arrives at node j exactly at time $t + \delta_{i,j}$. Constraints (7) have many implications. Let A be the first sum on the left side of Constraints (7) taken over all possible arrivals at node i at time t from some node j and let B be the second sum on the left side of Constraints (7), taken over all possible departures from node i at time t to some node j. Because all the $a_{i,j,t}$ and $\ell_{i,j,t}$ variables fact binary, we have $A \geq 0$ and $B \geq 0$. Because $z_{i,t} \leq 1$, these constraints ensure that $A + B \leq 1$. Therefore, for any node i and time t, the IP can select at most one destination j for a departure from i or at most one origin j for an arrival at i, and not both. Thus, Constraints (7) ensure that once the AUV arrives at j it cannot leave immediately, since $A = 1$ forces $z_{i,t} = 1$, and there cannot be both an arrival and a departure at the same time. Also, to leave a node, the AUV must be at that node. That is, if $z_{i,t} = 0$, indicating the AUV is not at node i at time t, then we have $A + B = 0$ so there can be neither a departure from nor an arrival at node i during time t.

Constraints (8) ensure that the $z_{i,t}$ variables correctly track the AUV through time. The first sum on the right hand side is the same A described above for Constraints(7) and the second (subtracted) sum is the same as B. By Constraints(7), each of A and B is binary and they cannot both be 1. Thus $-1 \leq A - B \leq 1$. We also have that $z_{i,t-1}$ and A cannot both be 1. This is because if $A = 1$, then $B = 0$, so if $z_{i,t-1} = 1$ as well, Constraint (8) would give $z_{i,t} = 2$, which is impossible for binary $z_{i,t}$. There are only four possible ways to set the $(z_{i,t}, z_{i,t-1}, A, B)$ tuple. We can now interpret Constraints (8). If there are no arrivals or departures ($A = B = 0$), then $z_{i,t} = z_{i,t-1}$, so the AUV stays at node i if it was there at time $t-1$ and otherwise, it is there at neither time. If there is an arrival ($A = 1$), then we have $B = z_{i,t-1} = 0$ from the above arguments, and the constraint sets $z_{i,t} = 1$. So an arrival at node i at time t places the AUV at node i at time t. If there is a departure ($B = 1$), then we have $A = 0$ from the above arguments. We must then have $z_{i,t-1} = 1$ to keep the right side of the constraint non-negative, which sets $z_{i,t} = 0$. That is, if the AUV is at node i at time $t-1$ and it leaves node i at time $t-1$, then it is not at node i at time t.

Constraints (9) are a special case of Constraints (8) for the first time τ_i when the AUV can arrive at a sensor node i. Because the variables z_{it} are only defined for $t \in L(i)$, the set of times the AUV can be at node i, and τ_i is the first such time, then variable $z_{i,t-1}$ is not defined for time τ_i. So there are no Constraints (8) for sensor nodes i at time τ_i. Constraints (9) say that the AUV is at node i at time τ_i if and only if the AUV left the surface from a closest surface node j at time 0 headed to node i. Any later arrivals are covered by Constraints (8).

Constraints (10) ensure that the AUV is on the surface while it delivers any data chunk. If a data chunk is delivered to the sink at time t, then the AUV sends the data to the sink during time $t-1$. Constraints (14) ensure the AUV is at the surface at time $t-1$ (see below). However, we must also ensure the AUV stays all the way through to time t. Constraints (8) might otherwise allow the AUV to leave at time $t-1$.

Constraints (11) ensure that the AUV is at node i during the time when it collects any data chunk from node i. Data collected during time $t-1$ is considered fully collected at time t. Constraints (15) ensure the AUV is at node i at the start of slot $t-1$ if the AUV collects a data chunk during time $t-1$. Constraints (11) ensure the AUV stays all the way through to time t.

Constraints (12) ensure that before data chunk $d_{i,p}$ is delivered to the sink it has been collected from node i at a legal time: after it is released and with enough time for the AUV to bring it to the surface.

Constraints (13) ensure that a data chunk is given credit for delivery at most once. Otherwise, the IP might "cheat" and claim delivery of the same chunk at two or more times.

Constraints (14) ensure that the data chunks broadcast to the sink during time $t-1$ (arriving at time t) require no more than one time unit to send. That is, this enforces constraints on the broadcast capacity of the AUV-to-sink connection. Constraints (15) similarly enforce capacity constraints on the AUV-to-sensor communication: the total time to collect all the data chunks scheduled for collection at time $t-1$ is no more than one time unit. These constraints also ensure that the AUV is at the surface (for broadcast to the sink) or at node i (for collecting from node i). If the right-hand-side AUV location variable is 0 in either type of constraint, then there can be no broadcast or collection variables turned on in the left side Constraints (14) and (15), respectively.

The remaining constraints concern the domain and definition of the variables:

4 Greedy Heuristic

Solving the ILP described above gives us the complete, optimal path of the AUV, but takes as input the VoI of the data chunks, effectively requiring us to have advance knowledge of the events and their values. In this section, we describe a path finding strategy where the current plan is adapted to the occurrence of new events as the AUV travels through locations. Being adaptive and not requiring a priori knowledge of events, this strategy can be used in actual UWSNs. As before, the AUV moves from a given starting point on the surface, visits nodes to collect their data, emerges at known surfacing locations to offload them, and by the end of network operations (i.e., at time T), it returns to one of the surfacing locations.

When a new event occurs, the node sensing it generates a small *event packet* describing the value, type of decay and urgency of the recording. This packet is, of course, much too short to contain the actual recording. The event packet is transmitted to the AUV acoustically either through single hop communication or through a simple flooding mechanism (e.g., EFlood [13], [14]). The node then starts recording the event, and every ϑ time units stores a data chunk. Throughout the duration of the event, the sensing node keeps sending event packets to the AUV, thus keeping it informed on the monitoring of the event and on the production of data chunks.

The AUV follows a greedy strategy for visiting the nodes: At every decision point the next node to be visited will be the one that offers the greatest contribution to the VoI that will be delivered. The AUV can also possibly greedily change its path as it goes to a node if visiting another node would result in a higher delivered VoI. Because the approach to path finding follows a greedy strategy and adapts to the dynamic occurrence of events, we name our heuristic *GAAP*, for Greedy and Adaptive AUV Path finding.

In the following we describe the operations of GAAP in detail, starting from the AUV computing the maximum VoI attainable from visiting a node and concluding with the overall protocol operations. In what follows we assume that all time values are calculated in the reference timeframe of the AUV and that a node clock synchronizes to that of the AUV at every visit.[2]

The core component of GAAP is the calculation of the VoI that can be obtained by visiting a sensor node S_i.

The algorithm takes as input the information carried by the event packet that triggers it. This includes the node ID S_i, the time ϑ needed to produce a data chunk, information on the VoI of the data chunks (e.g., initial value, type of decay, deadline), and the end time T of network operations. The AUV also considers the predicted duration T_i of the event (chosen based on past history) and the time t_c when the AUV starts the computation of the VoI. The output of the algorithm is the following triple: The VoI $\mathcal{V}_{S_i}^{T_i}$ that the AUV can deliver to the sink for the event sensed at node S_i; the final time t_{f_i} at which all data chunks from S_i are delivered to the sink, and the collection and delivery strategy $\mathcal{S}^i$ that obtains $\mathcal{V}_{S_i}^{T_i}$.

Based on the input the AUV builds a list L_d of records of all the data chunks that will be produced by node S_i for the event it is sensing (line 1). The length $|L_d|$ of the list depends on the predicted duration T_i and on the time ϑ needed to generate the data chunk: $|L_d| = \lceil T_i/\vartheta \rceil$. The value of the VoI of each chunk described in L_d depends on the time t_c when the AUV starts the computation and on the time of arrival of the AUV at node S_i (easily computed, as the distance between the current position of the AUV and the node and the AUV speed are known).

Algorithm 1: *VoIFromNode*$(S_i, \vartheta, \text{VoI info}, T_i, t_c, T)$

```
   Output:
       $\mathcal{V}_{S_i}^{T_i}$: VoI from node $S_i$
       $t_{f_i}$: Delivery time of data chunks from node $S_i$
       $\mathcal{S}^i$: Strategy of collection and delivery
1  $L_d$ = VoI-based queue of data chunks info;
2  $\mathcal{V}_{S_i}^{T_i} = 0$;
3  $t_{f_i} = 0$;
4  for $\varphi \leftarrow 1$ to $|L_d|$ do
5      $VoI_\varphi = \sum$ VoI of data chunks delivered $\varphi$ at a time;
6      $t_\varphi$ = time it takes to collect and deliver all data
       chunks;
7      if $VoI_\varphi \geq \mathcal{V}_{S_i}^{T_i}$ then
8          $t_{f_i} = t_c + t_\varphi$;
9          if $t_{f_i} > T$ then
10             break;
11         $\mathcal{V}_{S_i}^{T_i} = VoI_\varphi$;
12         $\mathcal{S}^i$ = Deliver $\varphi$ data chunks at a time;
13 return $\langle \mathcal{V}_{S_i}^{T_i}, t_{f_i}, \mathcal{S}^i \rangle$;
```

In order to compute the highest deliverable VoI of data chunks from the event sensed by node S_i, the AUV compares the VoI of multiple collection and delivery strategies, which consist in collecting the data chunks from the node, surfacing to the closest surface location, offloading to the sink, and traveling back to the node. To avoid the combinatorial explosion of computing the VoI from all possible subsets of data chunks, algorithm *VoIFromNode* heuristically tries out simple strategies, consisting of computing the VoI of collecting and delivering one data chunk at a time, two data chunks at a time, etc., till $|L_d|$ data chunks at a time (main **for**; lines 4 to 12). The strategy $\mathcal{S}^i$ that obtains the highest VoI is kept and eventually returned, along with highest VoI $\mathcal{V}_{S_i}^{T_i}$ and the time t_{f_i} when the last data chunk of the event sensed by node S_i is delivered (line 13). We notice that if the delivery time t_{f_i} is greater than the time T marking the end of network operations, the algorithm exits the main **for**, outputting the VoI of the data chunks that can be delivered before the network stops. The computational complexity of algorithm *VoIFromNode* is $O(|L_d|)$, i.e., linear in the number of data chunks produced for the sensed event.

GAAP finds paths depending on a VoI-based value that the AUV assigns to each node that has sent an event packet: The *VoI-Score*. Specifically, the VoI-Score of node S_i is defined as the ratio of the expected VoI $\mathcal{V}_{S_i}^{T_i}$ obtained by delivering data chunks of node S_i (algorithm *VoIFromNode*) and the time $t_{f_i} - t_{now}$ needed to perform that delivery, where t_{now} is the time at which the AUV computes the score:

$$\text{VoI-Score of node } S_i = \frac{\mathcal{V}_{S_i}^{T_i}}{t_{f_i} - t_{now}}. \tag{22}$$

If the AUV has received event packets from both nodes S_i

[2] Compared to the travel time of the AUV between nodes clock drifts are negligible.

and S_j, it executes algorithm *CombinedVoI-Score* below to determine which node to visit first.

```
Algorithm 2: CombinedVoI-Score($S_i, T_i, S_j, T_j$)
Output:
    Node to be visited first
    VoI-Score of visiting $S_i$ first or $S_j$ first
1  $\vartheta$ = recording time of a data chunk;
2  $t_{now}$ = current time;
3  $\langle \mathcal{V}_{S_i}^{T_i}, t_{f_i}, \mathcal{S}^i \rangle$ = VoIFromNode($S_i, \vartheta$, VoI info, $T_i, t_{now}$);
4  $\langle \mathcal{V}_{S_j}^{T_j}, t_{f_j}, \mathcal{S}^j \rangle$ = VoIFromNode($S_j, \vartheta$, VoI info, $T_j, t_{f_i}$);
5  VoI-Score$_{S_i \to S_j} = \frac{\mathcal{V}_{S_i}^{T_i} + \mathcal{V}_{S_j}^{T_j}}{t_{f_j} - t_{now}}$;
6  $t_{now}$ = current time;
7  $\langle \mathcal{V}_{S_j}^{T_j}, t_{f_j}, \mathcal{S}^j \rangle$ = VoIFromNode($S_j, \vartheta$, VoI info, $T_j, t_{now}$);
8  $\langle \mathcal{V}_{S_i}^{T_i}, t_{f_i}, \mathcal{S}^i \rangle$ = VoIFromNode($S_i, \vartheta$, VoI info, $T_i, t_{f_j}$);
9  VoI-Score$_{S_j \to S_i} = \frac{\mathcal{V}_{S_i}^{T_i} + \mathcal{V}_{S_j}^{T_j}}{t_{f_i} - t_{now}}$;
10 if VoI-Score$_{S_i \to S_j}$ > VoI-Score$_{S_j \to S_i}$ then
11     return $\langle S_i$, VoI-Score$_{S_i \to S_j}, \mathcal{S}^i \rangle$;
12 else
13     return $\langle S_j$, VoI-Score$_{S_j \to S_i}, \mathcal{S}^j \rangle$;
```

The algorithm takes as input the IDs of nodes S_i and S_j and the predicted duration times T_i and T_j of their events. Then it uses algorithm *VoIFromNode* to compute the values $\mathcal{V}_{S_i}^{T_i}$ and $\mathcal{V}_{S_j}^{T_j}$ of the VoI that the AUV would deliver to the user by visiting node S_i first and then node S_j, respectively (lines 3–4). At this point, the combined VoI-Score of visiting node S_i before node S_j is determined (line 5). Lines 3 to 5 are repeated for the case when node S_j is visited first. The two combined VoI-Scores are compared (line 10) and the algorithm returns the node that, if visited first, would allow the AUV to deliver the data chunks with the highest VoI, the value of the highest VoI itself, and the collection and delivery strategy that determines it. The computational complexity of algorithm *CombinedVoI-Score* is the same of that of algorithm *VoIFromNode*.

The combined VoI-Score is used by the AUV to determine the node to visit next every time it has finished delivering the data chunks of a node. This is when the AUV considers all event packets that it has received and for each of their sources computes the VoI-Score by executing the algorithm *NodeSelection*.[3]

The algorithm takes as input all the information about the events currently being sensed in the network. Based on such information, repeatedly running algorithm *VoIFromNode*, it selects the node S_i whose data chunk would provide the highest VoI (lines 1 and 2). The AUV then computes the combined VoI-Score of all possible pairs $\langle S_i, S_j \rangle$ (lines 4 and 5). The node S_k that will be visited next is the one that obtains the highest VoI-Score (line 7). The computational complexity of algorithm *NodeSelection* is $O(|\mathcal{E}||L_d^{max}|)$, where L_d^{max} is the longest list of records of data chunks among those of all the sensing nodes in $\mathcal{E}$.

```
Algorithm 3: NodeSelection(VoI Info)
Output:
    Node $S_k$ to be visited
1  $\mathcal{E}$ = Set of nodes sensing an event;
2  $S_i$ = Node in $\mathcal{E}$ with the highest $\mathcal{V}_{S_i}^{T_i}$;
3  $\langle S_k$, score$_k \rangle = \langle 0, 0 \rangle$;
4  for $S_j \in \mathcal{E}$: $S_j \neq S_i$ do
5      $\langle S_{i|j}$, score$_{i|j} \rangle$ = CombinedVoI-Score($S_i, T_i, S_j, T_j$);
6      if score$_{i|j}$ > score$_k$ then
7          $\langle S_k$, score$_k \rangle = \langle S_{i|j}$, score$_{i|j} \rangle$
8  return $S_k$;
```

As an example, let us consider the case of the AUV on the surface with three event packets from nodes S_i, S_j and S_k. Through Equation (22), the AUV computes the VoI-Score of the three nodes. Let node S_i be the node with the highest score, say, VoI-Score of node $S_i = 12$. The AUV now runs algorithm *CombinedVoI-Score* twice, with input S_i and S_j first and then with input S_i and S_k. Let us assume that the outputs of the two runs are $\langle S_i, 13, \mathcal{S}^i \rangle$ and $\langle S_k, 17, \mathcal{S}^k \rangle$, respectively. The AUV will start by visiting node S_k and collecting and delivering its data chunks according to the strategy $\mathcal{S}^k$. After having visited node S_k and collected and delivered its data chunks, the AUV decides how to proceed next based on stored event packets, including those from node S_i and S_j that were not visited earlier and newly received ones.

The reason why we use the combined VoI-Score of sequential visits to pairs of nodes rather than greedily visiting the node with the highest VoI-Score is to take into account the decay types and deadlines of the VoI of different events. For example, let us consider a scenario with a node S_i sensing an event whose VoI does not decay and a node S_j sensing an event whose VoI decays exponentially. If the two events have similar VoI values and deadlines, the node with the highest VoI-Score would surely be node S_i. A simple greedy strategy would favor visiting this node first. For sufficiently long deadlines, however, it could be best to visit node S_j first, and then node S_i, thus clearly delivering a higher final VoI. This situation is captured by the notion of combined VoI-Score and its computation by algorithm *CombinedVoI-Score*.

In scenarios where the AUV spends a considerable amount of time traveling between nodes it might happen that it receives a high number of event packets. In order to decrease the likelihood of missing events with high VoI, we allow the AUV to consider changing its path as it travels to a node. For instance, let us consider the case of the AUV traveling to node S_i and receiving $p \geq 1$ event packets of new events from p other nodes S_j. In this case, the AUV immediately runs algorithm *CombinedVoI-Score* p times with input S_i and S_j and the predicted times of their events. The node resulting from algorithm *CombinedVoI-Score* as the node with the highest combined VoI-Score is the one eventually visited. Once its data chunks are collected and delivered, path finding proceeds as described above.

5 Performance Evaluation

We present the results of a simulation-based performance evaluation of the solutions proposed in this paper. We start the

[3] The AUV will not interrupt data collection from a node or traveling to the surface to deliver data chunks. Event packets arriving at these times will be stored and processed as soon as the AUV is done.

section by introducing the simulation scenarios and their parameters. We then show results on the performance of GAAP with respect to the upper bound (termed OPT) provided by solving the ILP model (Section 5.2). Finally, we present a comparative performance evaluation of GAAP and several heuristics for AUV path finding aiming at showing the impact of VoI-aware mobility (Section 5.3). All our results concern the total VoI that the AUV delivers to the user by the end of network operations. We also compare GAAP and other AUV path finding heuristics with respect to other metrics, namely, the energy consumed for information transfers and for moving the AUV throughout the network normalized to the delivered VoI ("energy efficiency"). For the sake of conciseness, in the following we write "GAAP delivers the VoI . . . " to intend "the AUV moving according to GAAP delivers data chunks whose VoI . . . ," and similarly for OPT and for the other considered heuristics.

5.1 Simulators and simulation scenarios

The results for OPT have been obtained by solving the ILP model defined in Section 3 using Pyomo [15] and the freely available software Gurobi [16] run on Linux-based 64-bit multiple-core servers (we used Gurobi default settings). Each of the three servers we used has 16 cores, clocked at 2.8GHz, and 64GB of RAM. The various runs took from several hours to a few days to produce the optimal solutions. We implemented GAAP (Section 4) in a home-grown software framework and simulated network communications in SUNSET, a framework for underwater emulation/simulation [17].

All our experiments consider realistic parameters of UWSNs. We consider topologies with $|S| = 4$, 5, 9, 12, 18 and 35 wireless underwater sensor nodes deployed over a rectangular area 2km×3km. Nodes are deployed at a depth chosen at random between 50m and 200m. Fig. 2 shows the layout of the topologies we considered as seen from above.

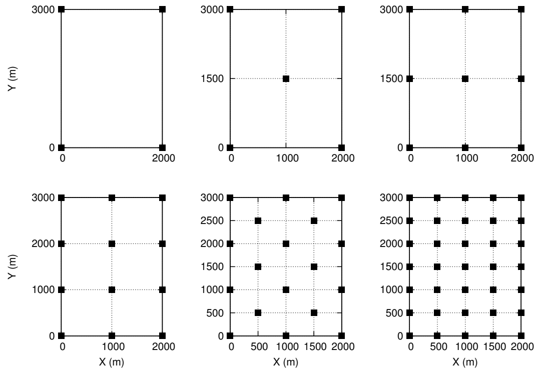


Fig. 2: The topologies considered in our experiments. Nodes are deployed in a grid-like fashion over a 2km×3km area.

Our scenarios include $|S|$ surfacing points, located directly above each of the $|S|$ nodes. We set the AUV cruise speed to 1.8m/s, to match that of a Remus class vehicle [18].

Each node sends event packets to the AUV over the acoustic data channel. The packet size is set to 10B (which includes the headers of all protocol layers, from network down to physical). The acoustic channel data rate is set to 4Kbps. When direct communication between node and AUV is not possible, event packets are disseminated by using the flooding algorithm implemented in SUNSET (E-Flooding). For these communications the modem transmission power is set to 3.3W and its reception power is set to 0.5W. We set the optical and wireless data transfer rate to 10Mbps. Optical communication can reach this data rate when the AUV hovers within 100m of a node [3], [4]. The muddier the water, the closer the AUV should be to the node for transfer to be successful. Power consumption for optical data exchange is set to 3W [2]. Delays of event packet notifications to the AUV are computed and added to the AUV traveling delays for the final computation of the VoI of the collected data chunks.

We consider a scenario where the nodes use cameras to take videos (e.g., for monitoring and/or intrusion detection). Surveillance data are stored as 720p high-definition videos, with a resolution of 1280 × 720 pixels at 3 frames/s.

Event arrival is modeled by a Poisson process with arrival rate $\lambda = 1$ hour. Once generated, an event is assigned a location randomly and uniformly within the 3D network deployment area. It is also assigned a random duration that is exponentially distributed, with 1 hour average. Events have an initial VoI varying between 0.4 (a non important event, like nothing is happening) and a maximum value depending on the event, varying in the set $\{20, 50, 100, 200\}$ (a very important event like the detection of an intruder). The actual value perceived and reported by the sensing node depends on the distance from the node and the location of the event according to a Gaussian-like distribution. Fig. 3 shows the reported VoI of an event whose value is 20 depending on its distance from its sensing node.

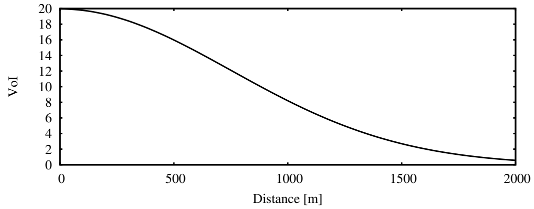


Fig. 3: The initial VoI of a data chunk depends on the distance between the location of the event and that of the node sensing it according to a Gaussian-like distribution.

An event that happens a few meters from a node will be announced to the AUV with its full value. If the event happens farther away from the node, its announced value will be lower, according to the selected distribution curve. For example, at 1000m from a node the VoI of each data chunk produced for an event whose full VoI is 20 will be 9 (Fig. 3). We stipulate that the event is sensed, and therefore recorded and reported, by only one node, namely, the one geographically closest to the event location. (Unlikely ties are broken by using the nodes unique ID.) Being the closest to the event this node will produce data chunks with the highest initial VoI.

The VoI of produced data chunks decays according to two different functions: An exponential decay and a "non decay" function. The first function is for data chunks that need to delivered as soon as possible. Less sensitive data, whose value does not decrease in time are given a VoI that does not decay. Fig. 4 shows different decays for the VoI of a data chunk, corresponding to different initial VoI. The figure also shows that after a given deadline the VoI of a data chunk goes to 0, signifying that there is no longer an interest to deliver that data chunk.

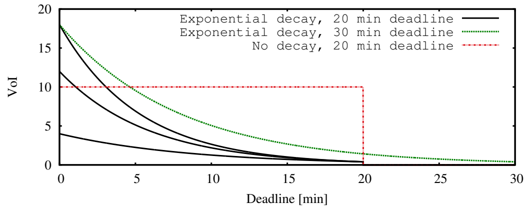


Fig. 4: The different types of decay for the VoI of a data chunk depending on different initial VoI. After a given deadline the VoI of a data chunk goes to 0 (VoI deadline).

We set the duration ϑ of the recording of a data chunk to 5 minutes.[4] For instance, for an event lasting 23 minutes, the sensing node will generate 5 data chunks, the first four of which will last the full ϑ, and the last one only 3 minutes. Assuming a video encoded using the standard H.264 codec, a 5 minute recording produces a 9MB data chunk. The duration of network operation T is set to 12 hours. We consider time units of 1 minute each.

5.2 GAAP vs. OPT

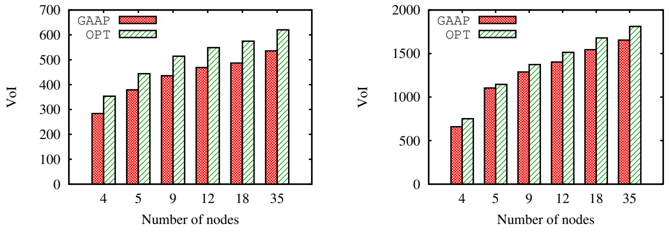
We compare the performance of GAAP and of the optimal benchmark OPT with respect to the total VoI of the data chunks delivered to the user by the end of network operations. We start with showing results for the case of *uniform decay*, namely, of when the VoI of all data chunks decays either exponentially or does not decay. We then consider the case of *heterogeneous decay*, where data chunks of half of the events have a VoI that decays exponentially, while the VoI of the data chunks of the other half does not decay. For this set of experiments the full value of the VoI of all data chunks is set to 20.

Our results are obtained by averaging over 30 experiments for both GAAP and OPT. In each experiment we randomly and uniformly vary the set of the events and their location. In the experiments concerning GAAP we set the AUV starting point as one of the surface locations closer to the center of the (surface of the) deployment area. In the scenario with 4 nodes the starting point is chosen randomly and uniformly among one of the 4 surface corner locations.

GAAP vs. OPT: Uniform decay

In these experiments we set the deadline for the VoI of the data chunks to 20 minutes, which is less than the 30 minutes it would take the AUV to travel the maximum distance in the deployment area. This is to show how GAAP adapts to events the VoI of whose data chunks expires before the AUV can even get to their source node. In this case it would not make sense for the AUV to go pick up and deliver those data chunks.

Fig. 5 shows the VoI of data chunks delivered in networks with increasing number of nodes. As expected, OPT outperforms GAAP because of the centralized, all-knowing nature of ILP modeling. The gap between the two, however, is reasonably low, independently of the size of the network, the type of decay, and of the data chunk recording time. In particular, in the case where the VoI decays exponentially (Fig. 5a), we observe that GAAP delivers a total VoI that is at most 20% lower than that of OPT. This happens in networks with 4 nodes, as the AUV has to travel larger distances, and, given the exponential decay, by the time the AUV arrives at a corner node the VoI of some data chunks has reached the deadline, i.e., it is 0. OPT knows already where and when an important event is going to happen, and sends the AUV at its sensing node in advance. In denser networks, where more and more nodes are placed towards the center of the deployment area, events are detected by these nodes, and the distances traveled by the AUV become increasingly shorter. This decreases the gap between GAAP and OPT, which is already 16% in the case of networks with 5 nodes, and goes down to 14% in highly dense networks ($|S| = 35$).



(a) Exponential decay

(b) No decay

Fig. 5: GAAP vs. OPT: VoI of data chunks delivered in networks with increasing number of nodes. Independently of the type of decay GAAP delivers a total VoI that is at most 20% lower than that of OPT.

Fig. 5b concerns the case when the VoI of data chunks does not decay for 20 minutes. The total VoI is considerably higher, since as far as the AUV gets to collect and deliver data chunks before their deadline, their VoI stays the same. The gap between OPT and GAAP is also significantly reduced, being GAAP less penalized by the fact that the AUV needs to be made aware of events rather than knowing that information in advance as it is for OPT. The total VoI delivered by GAAP is only 13% less than that from OPT in the worst case of networks with 4 nodes. This difference goes down to 9% in networks with 35 nodes.

GAAP vs. OPT: Heterogeneous decays

The experiments of this section concern scenarios where half of the events that happen in the network are recorded by data chunks whose VoI decays exponentially, while the VoI of data chunks of the other half of the events does not decay. We set the deadline of data chunks from the first kind of events to 30 minutes, and that of the “no decay” events to 60 minutes. With this choice we intend to explore the effectiveness of using algorithm *CombinedVoI-Score* (Section 4), through which we aim at sending the AUV first to a node whose data chunk VoI decays exponentially and then to a node whose data chunk VoI does not decay (till a deadline), if that eventually produces a higher VoI. To show the “power” of algorithm *CombinedVoI-Score*, GAAP is not only compared to OPT, but also to a “myopic” version of itself, termed GAAP-M, where instead of

[4] We have performed experiments with $\vartheta = 1$ and $\vartheta = 10$ and observed similar trends, with variations concerning only the actual value of the delivered VoI.

choosing a path based on the combined VoI-Score, the choice of which node to visit first is solely based on the VoI-Score. In case the deadline of the two different types of decay was set to be the same, GAAP-M would always send the AUV to the node sensing an event whose data chunks have VoI that does not decay.

Results for GAAP, GAAP-M and OPT are shown in Fig. 6.

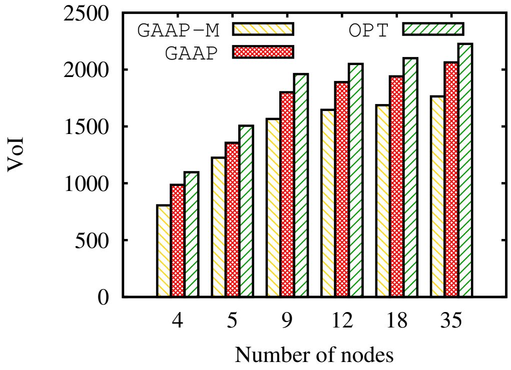
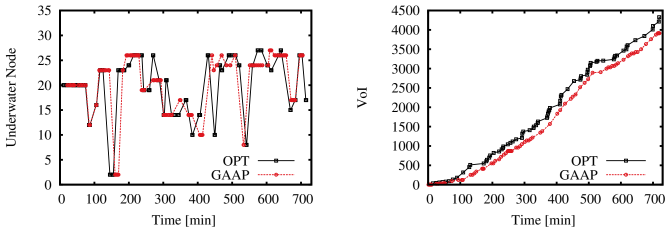


Fig. 6: GAAP vs. OPT: VoI of data chunks decaying exponentially or not decaying. GAAP delivers a VoI that is at most 11% lower than that delivered by OPT. The performance gap between the two decreases monotonously as the network size increases.

We observe that GAAP delivers a VoI that is at most 11% lower than that delivered by OPT. This happens for the sparsest networks we consider. As noticed in the case of scenarios with uniform decay, as the network density increases, the gap between GAAP and the benchmark decreases monotonously to 8%, which happens when $|S| = 35$.

The effectiveness of using the combined VoI-Score, i.e., of algorithm *CombinedVoI-Score*, is shown by the performance of GAAP-M. GAAP-M always delivers a VoI that is at least 20% lower than that delivered by GAAP.

We end this section by showing results that demonstrate how GAAP operations mimic quite closely the operations of OPT (Fig. 7). These results refer to the scenario of one network with 35 nodes (numbered 1 through 35), with heterogeneous decay, and with a larger number of events than in the scenarios considered above (the Poisson process that models event arrival has a rate of 30 minutes).



(a) Positions over time

(b) VoI over time

Fig. 7: The path followed by OPT and GAAP (a), and the VoI delivered over time (b). GAAP visits 90% of the nodes visited by OPT and delivers a VoI consistently close to the optimum.

Fig. 7a shows the actual path (as the sequence of visited nodes) followed by the AUV as driven by GAAP and OPT, respectively, over the 720 minutes of network operations. We notice that GAAP visits 90%, i.e., almost all, of the nodes visited by OPT, and almost in the same sequence. This is why, as shown in Fig. 7b, the VoI delivered by GAAP and OPT over time shows the same trend. These results provide further evidence of the effectiveness of the design of GAAP, and of the results shown so far.

5.3 Impact of VoI awareness

Our second set of experiments is aimed at quantifying the impact of VoI awareness on the performance of path finding protocols for AUVs. To this purpose we present results from a comparative performance evaluation of GAAP and three non VoI-aware path finding strategies. The strategies that we consider are all "event-aware," in the sense that the AUV knows which nodes are sensing an event (as for GAAP, we use event packets). In every heuristic the AUV computes the strategy $\mathcal{S}^i$ for collecting and delivering data chunks along the lines of algorithm *VoIFromNode* (Section 4), namely, determining the number φ of data chunks to collect and deliver at a time that produces the highest VoI. The selected path finding strategies are the following.

Random Selection (RS). The AUV travels to a node S_i chosen randomly and uniformly among those that are sensing an event. Once at a node, it collects all its data chunks, resurfaces and delivers them according to strategy $\mathcal{S}^i$. Once done, a new nodes is selected randomly and uniformly among those that are currently sensing an event, an so on.

TSP. Before the start of network operations, a traveling salesman problem (TSP) path is found off line based on internodal distances.[5] The AUV starts from a surface point (the same used by GAAP) and visits the closest node S_i in the TSP path that is sensing an event. Once at a node the AUV collects all its data chunks, resurfaces and delivers them using strategy $\mathcal{S}^i$. When done, the AUV proceeds to visit the next node in the TSP path that is sensing an event, and so on.[6]

Lawn Mower (LM). A path is found off-line following a so-called lawn mower trajectory. This path is found once and for all before the start of network operations. According to this strategy, the network nodes of the considered network topologies (Fig. 2) are visited line by line, left to right, from bottom to top, and back. As such, this heuristic allows some nodes to be visited more than others. For instance, in a network with 5 nodes, the AUV would visit first the two on the bottom line (left one first), then the single central node, then the two node in the top line (left one first), and then back to the single central node, before moving back to the first node. In other words, before starting the journey again from the first node, the central node has been visited twice. The AUV starts from a surface point (the same used by GAAP) and visits the closest node in the LM path that is sensing an event. As for the previous strategies, once at a node S_i the AUV collects all its data chunks, resurfaces and delivers them according to strategy $\mathcal{S}^i$. Once done, the AUV proceeds to visit the next node in the LM path that is sensing an event, and so on.

[5] In our experiments, given a network topology, we found a TSP path by using the free Concorde TSP solver [19].

[6] Note that if a node is not sensing an event, even if it is next in the TSP path, is skipped, as it would make no sense to visit a node that has no data chunk, i.e., that is not producing information of any value.

Investigated metrics

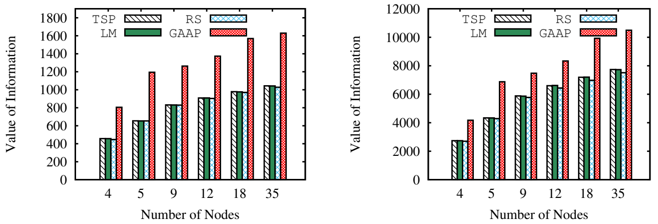
All protocols are compared with respect to the following metrics.

- *Delivered VoI*, which is the sum of the VoI of all data chunks delivered to the sink throughout the time of network operations.
- *Energy efficiency*, defined as the ratio between the total energy consumption and the delivered VoI. The total energy consumption is obtained by adding the energy needed for data chunk transfer (optical), the energy consumed for flooding event packets to the AUV (acoustic), as well as that consumed by the AUV to move (we used the Remus vehicle energy consumption model [18]).
- *Throughput*, defined as total number of bytes delivered to the sink over the time of network operations.
- *End-to-end latency*, which is the time from when a data chunk is generated to when it is delivered to the sink.

For this set of experiments we consider events recorded by data chunks with uniform decays (both exponential and no decay) with the deadline set to 20 minutes. All results are obtained by averaging over data from 500 simulation runs, which achieves a statistical confidence of 95% within a 5% precision.

Performance results: Delivered VoI

Fig. 8 shows results about the delivered VoI. When the VoI decays exponentially (Fig. 8a) the VoI delivered by GAAP in networks with 4 nodes is at least 77% more that that of TSP and LM, and 80% more than RS, which we observed to be always the worst strategy among those considered. As the density of the network increases the VoI delivered by all strategies increases as events happen closer to the nodes and the VoI of their data chunk is higher (Fig. 3). Furthermore, the AUV has to travel less between nodes, which incurs lower decays. Again, the VoI-aware behavior of GAAP allows the AUV to gather and deliver data chunks with higher value. For instance, in networks with 35 nodes, GAAP delivers 57% more VoI than TSP, which results to be the best among all other strategies.



(a) Exponential decay

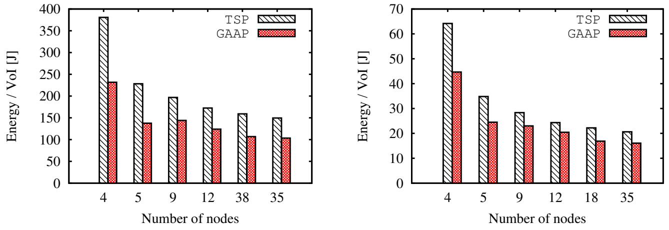
(b) No decay

Fig. 8: GAAP vs. other strategies: Independently of network size and type of decay GAAP delivers at least 40% more VoI than the best of the other solutions.

When the information does not decay for 20 minutes (Fig. 8b) the total VoI delivered to the user is higher for all strategies, as expected. GAAP still delivers 54% more VoI than the best among the other strategies (TSP) in sparse network (4 nodes). This betterment decreases to 40% more VoI in dense networks, where all strategies gain from nodes being closer, i.e., from having the AUV traveling lower distances.

Performance results: Energy efficiency

Results concerning energy efficiency are shown in Fig. 9.



(a) Exponential decay

(b) No decay

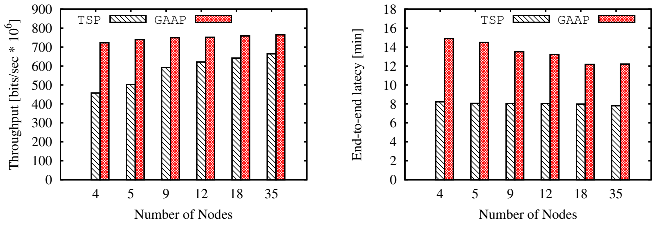
Fig. 9: GAAP vs. other strategies: Independently of network size and type of decay GAAP is at least 30% energetically more efficient that TSP.

We show results only for GAAP and TSP, as the latter always outperforms RS and LM. We observe that the 97% of the energy for the overall system to work is required for moving the AUV. Independently of the decay of the VoI, the energy consumption per delivered VoI decreases as the network size increases because we observed that the total energy required by the AUV does not significantly vary but the delivered VoI increases (see above).

Fig. 9a concerns the energy efficiency of the two strategies when the VoI of data chunks decays exponentially. We notice that GAAP is about 70% more efficient than TSP in sparse networks. The gap between the two protocols decreases to 45% in the densest networks ($|S| = 35$). In the case of non decaying VoI (Fig. 9b) the absolute value of delivered VoI increases, which makes the energy required to deliver one unit of VoI decrease. GAAP is still the most efficient of the two strategies, resulting 46% more efficient than TSP in sparse networks ($|S| = 4$) and 30% more efficient in the densest case ($|S| = 35$).

Performance results: Throughput and end-to-end latency

Results for traditional networking metrics that are independent of the VoI delivered to the sink are shown in Fig. 10. (We report results only for GAAP and TSP, as the latter always outperforms RS and LM.)



(a) Throughput

(b) End-to-end latency

Fig. 10: GAAP vs. TSP: Throughput and end-to-end latency.

Throughput. The number of bits per second delivered by GAAP is always noticeably higher than that of all other heuristics (Fig. 10a). In particular, GAAP delivers about 60% more bits per second in sparse networks (with 4 or 5 nodes), about 23% more bits per second in networks with 9 to 12 nodes, and about 18% more bits per second in dense networks. This is because either GAAP and TSP visit nodes with data chunks in the same sequence, and in this case they collect the same number of bits, or GAAP arrives faster to nodes with new events, being able to collect more data chunks, i.e., more, and more valuable, bits. In the latter case, TSP will arrive to those nodes after having had to stop at other nodes, being able to collect only those data chunks whose VoI did not expire already. The advantage of GAAP over TSP decreases with the network size, as the travel time between nodes decreases, which enables TSP to collect more packets.

End-to-end latency. The latency experienced by data chunks delivered by GAAP is consistently higher than that of TSP (Fig. 10b). Latency almost doubles in sparse networks, it is 69% more in networks with 9 to 12 nodes, and it takes an average of 57% more time to deliver data chunks in dense networks. This result is consistent with our previous results on throughput as GAAP delivers a higher number of data chunks than TSP.

6 Related Works

Works showing the beneficial effects of combining optical and acoustic communications for underwater data retrieval and routing include [8], [9] and [7]. Specifically, in [8] Farr et al. describe and demonstrate a system with integrated optical and acoustic capabilities aimed at showing what data rates and ranges can be obtained by such a communication system. In a follow-up paper, Farr et al. show how the optical capabilities of the proposed integrated system greatly expands the bandwidth of a CORK seafloor borehole observatory [9]. This paper reports results from experiments where a surface vessel is sent to offload data from a CORK installation. A tethered probe with optical communication capabilities is lowered from the vessel to hover above the CORK optical telemetry system (OTS) and transfer data. The OTS is kept in a low power sleep mode, capable of listening for an acoustic wake-up request to prepare for data offload. The authors report data rates of up to 10Mbps, which enables a dramatic increases of the CORK sensor sampling frequencies and data transfer rates with respect to when only acoustic communications are used. These papers concern no data routing or data-driven path finding for vessels or AUVs to offload data from the nodes. Multi-hop routing for UWSNs with multi-modal communicating devices is the concern of the work of Hu and Fei [7]. Their protocol, called MURAO for MUlti-level Routing protocol for Acoustic-Optical UWSNs, partitions the network nodes into two layers. Lower layer nodes are responsible for actual data retrieval via hop-by-hop routing over optical channels. Nodes in the upper layer use long range/low bandwidth acoustic communication to coordinate the routing of the lower level nodes. This solution requires nodes to be deployed densely enough to obtain a connected topology over the optical links. Given the short range of the latter, MURAO can be costly and even impracticable for applications requiring networking large areas. Recently, Basagni et al. showed that machine learning-based decisions on communication mode and next-hop relay produce remarkable performance of multi-hop routing [20]. This work, however, considers only nodes with different acoustic modems, i.e., does not concern the interaction between optical and acoustic communications.

In scenarios including an AUV, new data retrieval methods can be defined in which underwater networking is enabled by device mobility, similarly to the scenario considered in our work. The mobile AUVs collect data from the nodes via short range optical links and then offload the data either by returning to the base station or surfacing and using over-the-air wireless communication. In this kind of UWSNs acoustic communication can be used for AUVs coordination or low bandwidth data transmission, such as signaling and control. Two of the earliest examples of this form of UWSNs are described by Vasilescu et al. [21] and Detweiller et al. [22]. Acoustic and optical communication protocols are integrated into the TinyOS operating system of sensor nodes and mobile AUVs. In [21] the results from a series of experiments is described that demonstrate the feasibility of data retrieval from static nodes visited by an AUV. It is assumed that nodes produce data of equal, non-decaying value at a constant rate, and that no data is lost by a node from overflow. In this case, AUV path finding is simply reduced to a sequential visit of the nodes one after another, according to a route pre-loaded to the AUV. The experiments described in [22] follow the same pre-determined AUV path strategy, and are primarily concerned with showing robust ranging between the mobile node and the static ones. Campagnaro et al. present a simulation-based study of multi-modal communication where an AUV periodically visits the 8 nodes of a UWSN following a pre-determined clockwise way-point trajectory [6]. Nodes switch between optical to acoustic communication in order to achieve the highest throughput, depending on channel conditions and the distance from the AUV. The implementation of the multi-modal networks stack used in the considered scenarios is presented in details in [5], where its functionalities are further demonstrated via simulations in a scenario comprising divers and a remotely operated underwater vehicle. While showing the effectiveness of combining acoustic and optical technologies for AUV-enabled underwater networking, these papers are not concerned with performance enhancement via AUV path finding optimization.

More recently, dynamic path finding for AUV-based data collection on multi-modal UWSN has been shown to be key to improve network performance from many points of view. Forero et al. investigate path finding for multiple AUVs considering constraints on energy, data storage and retrieval requirements [23]. A number of AUVs visits the UWSN nodes and exchange data optically. When an AUV available energy or its data storage capability are exhausting the AUV is sent to one among many designated depots to offload their data and re-charge. The depots, that are resource rich, are in charge to deliver data to the final user and have capabilities to recharge the AUVs. Determining AUV paths to the nodes and to the depots is the job of a Fusion Center (FC) with which the AUVs and the nodes communicate acoustically to send their current status. Optimal routing is centrally determined by the FC that keeps running a one-step lookahead with rollout algorithm and sends its output to the AUVs. Beyond being centralized and scarcely scalable, we notice that the routes that the FC determines for the AUVs do not depend on any valued associated to the sensed data.

One of the characteristics of the UWSN scenario is that the data rate of modern sensors greatly exceeds the capabilities of underwater networking channels. It is simply impossible to transfer all the sensed data in near-real time. Therefore, if the value of a data chunk varies in time, the system needs to make decisions about which data chunks are to be transferred preferentially. We argue that the intellectual framework of *Value of Information* (VoI) as considered here is an appropriate, disciplined way to make these decisions. The concept of VoI was originally proposed in game theory [24]. The intuition behind the game theoretical definition of VoI is the price an optimal player would pay for a piece of information. In the context of wireless sensor networks, a number of recent projects have introduced metrics to model situations where one either needs to select a subset of the collected data or choose between transmitting a piece of information or not. Bisdikian et al. [25], [26] define "Quality of Information" (QoI) as the degree to which a piece of information is (or is perceived to be) fit-to-use for a particular purpose. The QoI is usually conceived as a vector of quality attributes that include accuracy, latency, and spatio-temporal relevance. In the realm of UWSNs, one of the first works concerned with the problem of finding paths for an AUV that consider the quality of the information to be retrieved is the paper by Hollinger et al. [27]. In the scenario considered in the paper, communications between the nodes and the AUV happens only via acoustic communication. The AUV path finding problem is formulated as a variation of TSP where path cost (e.g., traveling time and fuel expenditure) and the information quality of the collected data are joint optimized. Information quality is here defined as the expected information gain at the AUV going through the found path, i.e., in terms of the number of correctly received packets carrying distinct information, which is probabilistically dependent on the quality of the acoustic channel. Therefore, the quality of information does not depend on the event recorded by the sensors, and does not decay in time. Turgut et al. [28], [29] define "pragmatic VoI" as the support the information gives to the decisions and actions of the network operator (without assuming an optimal decision-maker). In a UWSN setting similar to the one considered in this paper, Bölöni et al. discuss a scenario where the VoI is used to schedule direct acoustic transmission of digests of large data chunks to the sink and the optical transfer of the whole chunk to an AUV moving according to a fixed trajectory [11]. This paper considers the decay of the VoI in time (in form of an exponential decay function), the fraction of VoI that is retained by digests of the original data, and introduces the concept of conditional VoI, which captures the novelty of a data chunk in the context of previously transferred data. Its contribution mainly concerns scheduling between acoustic and optical transmissions so that the VoI is maximized, without tackling the problem of finding paths for an AUV to deliver high VoI data.

7 Conclusions

We presented a mathematical model (OPT) and a greedy heuristic (GAAP) for driving an AUV to collect and deliver data with decaying value from nodes of a UWSN. The aim is to find paths for the AUV that maximize the Value of Information of the data delivered to the sink. Our ILP model considers realistic and desirable network sizes, data communication rates, distances and surfacing constraints. GAAP successfully mimics the optimal paths found by the all-knowing OPT and obtains VoI of the delivered data that is at most 20% lower than that obtained by the ILP model, as shown by experiments over networks with increasing number of nodes. The performance of GAAP has been compared to that of other path finding strategies, where sensing nodes are visited randomly (RS), according to a TSP tour or to a Lawn Mower (LM) strategy. Our results show that VoI-aware AUV mobility produces higher performance. In particular, GAAP delivers up to 77% more VoI than that delivered by the best of the other heuristics (TSP) and achieves an overall energy efficiency that is up to 70% better than that of other solutions.

References

[1] J. Heidemann, M. Stojanovic, and M. Zorzi. Underwater sensor networks: Applications, advances and challenges. *Philosophical Transactions of the Royal Society A*, 370:158–175, August 2 2012.

[2] D. Anguita, D. Brizzolara, and G. Parodi. Prospects and problems of optical diffuse wireless communication for underwater wireless sensor networks (UWSNs). In G. V. Merrett and Y. K. Tan, editors, *Wireless Sensor Networks: Application-Centric Design*, chapter 12, pages 275–300. InTech, December 14 2010.

[3] S. Harnon. Underwater optical wireless communication network. *Optical Engineering*, 49(1):1–6, January 2010.

[4] F. Hanson and S. Radic. High bandwidth underwater optical communication. *Applied Optics*, 47(2):277–283, 2008.

[5] F. Campagnaro, F. Guerra, P. Casari, R. Diamant, and M. Zorzi. Implementation of a multi-modal acoustic-optic underwater network protocol stack. In *Proceedings of MTS/IEEE OCEANS 2016*, pages 1–6, Shanghai, China, April 10–13 2016.

[6] F. Campagnaro, F. Favaro, F. Guerra, S. V. Calzado, M. Zorzi, and P. Casari. Simulation of multimodal optical and acoustic communications in underwater networks. In *Proceedings of the MTS/IEEE OCEANS 2015*, pages 1–6, Genova, Italy, May 18–21 2015.

[7] T. Hu and Y. Fei. MURAO: A multi-level routing protocol for acoustic-optical hybrid underwater wireless sensor networks. In *Proceedings of SECON 2012*, pages 218–226, Seul, Korea, June 18–21 2012.

[8] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey. An integrated, underwater optical/acoustic communications system. In *Proceedings of the MTS/IEEE OCEANS 2010*, pages 1–6, Sydney, Australia, May 24–27 2010.

[9] N. Farr, J. Ware, C. Pontbriand, T. Hammar, and M. Tivey. Optical communication system expands CORK seafloor observatory's bandwidth. In *Proceedings of the MTS/IEEE OCEANS 2010*, pages 1–6, Seattle, WA, September 20–23 2010.

[10] M. Doniec, C. Detweiler, I. Vasilescu, and D. Rus. Using optical communication for remote underwater robot operation. In *Proceedings of IEEE/RSJ Intelligent Robots and Systems, IROS 2010*, pages 4017–4022, Taipei, Taiwan, October 18–22 2010.

[11] L. Bölöni, D. Turgut, S. Basagni, and C. Petrioli. Scheduling data transmissions of underwater sensor nodes for maximizing value of information. In *Proceedings of IEEE Globecom 2013, Ad Hoc and Sensor Networking Symposium*, Atlanta, GA, December 9–13 2013.

[12] S. Basagni, L. Bölöni, P. Gjanci, C. Petrioli, C. A. Phillips, and D. Turgut. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In *Proceedings of IEEE Infocom 2014*, pages 988–996, Toronto, Canada, April 27–May 2 2014.

[13] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini. Channel replay-based performance evaluation of protocols for underwater routing. In *Proceedings of IEEE MTS/OCEANS 2014*, pages 1–7, Saint John, NL, Canada, September 14–19 2014.

[14] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini. CARP: A channel-aware routing protocol for underwater acoustic wireless networks. *Ad Hoc Networks*, 34:92–104, November 27 2015.

[15] W. E. Hart, C. Laird, J. P. Watson, and D. L. Woodruff. *Pyomo—Optimization Modeling in Python*. Springer Optimization and Its Applications. Springer, 2012.

[16] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2015. http://www.gurobi.com.

[17] C. Petrioli, R. Petroccia, J. R. Potter, and D. Spaccini. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Networks*, 34:224–238, 2015.

[18] R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, and A. Kukulya. Development of the REMUS 600 autonomous underwater vehicle. In *Proceedings of MTS/IEEE OCEANS 2005*, volume 2, pages 1301–1304, September 2005.

[19] Concorde TSP solver, 2005. http://www.tsp.gatech.edu/concorde.html.

[20] S. Basagni, V. Di Valerio, P. Gjanci, and C. Petrioli. Finding MARLIN: Exploiting multi-modal communications for reliable and low-latency underwater networking. In *Proceedings of IEEE Infocom 2017*, pages 1701–1709, Atlanta, GA, May 1–4 2017.

[21] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of ACM SenSys 2005*, pages 154–165, San Diego, CA, November 2–4 2005.

[22] C. Detweiller, I. Vasilescu, and D. Rus. An underwater sensor network with dual communications, sensing, and mobility. In *Proceedings of MST/IEEE OCEANS 2007—Europe*, pages 1–6, Aberdeen, Scotland, June 18–21 2007.

[23] P. A. Forero, S. K. Lapic, C. Wakayama, and M. Zorzi. Rollout algorithms for data storage-and energy-aware data retrieval using autonomous underwater vehicles. In *Proceedings of ACM WUWNet 2014*, pages 22:1–22:8, Rome, Italy, November 12–14 2014.

[24] R. A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

[25] C. Bisdikian, J. Branch, K. K. Leung, and R. I. Young. A letter soup for the quality of information in sensor networks. In *Proceedings of IQ2S 2009*, pages 1–6, Galveston, TX, March 9 2009.

[26] C. Bisdikian, L. M. Kaplan, M. B. Srivastava, D. J. Thornley, D. Verma, and R. I. Young. Building principles for a quality of information specification for sensor information. In *Proceedings of IEEE FUSION 2009*, pages 1370–1377, Seattle, WA, July 6–9 2009.

[27] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. S. Sukhatme, M. Stojanovic, H. Singh, and F. Hover. Underwater data collection using robotic sensor networks. *IEEE Journal on Selected Areas in Communications*, 30(5):899–911, 2012.

[28] D. Turgut and L. Bölöni. A pragmatic value of information approach for intruder tracking sensor networks. In *Proceedings of IEEE ICC 2012*, pages 4931–4936, Ottawa, Canada, June 10–15 2012.

[29] D. Turgut and L. Bölöni. IVE: Improving the value of information in energy-constrained intruder tracking sensor networks. In *Proceedings of IEEE ICC 2013*, pages 4953–4957, Budapest, Hungary, June 9–13 2013.

Petrika Gjanci holds a Ph.D. in computer science from the University of Rome "La Sapienza" (March 2017). He received his B.S. and M.S. degrees in computer engineering also from the University of Rome "La Sapienza," in 2009 and 2012, respectively. Dr. Gjanci's research interest concerns theoretical and experimental aspects of wireless sensor networking, with an emphasis on underwater communication and networking. He has collaborated extensively to large research projects including the EU FP 7 ICT project SUNRISE "Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet."

Chiara Petrioli received a Ph.D. in Computer Engineering from the University of Rome "La Sapienza" in 1998. She is currently a professor with the Computer Science department of "La Sapienza." Prof. Petrioli is director of the Sensor Networks and Embedded Systems laboratory (SENSES lab) and of the Cyber Physical System lab of "La Sapienza" center for Cyber Intelligence and Information Security. She is also founding partner of "La Sapienza" spinoff WSENSE, S.r.l. Her research interests focus on the design and optimization of wireless, embedded and cyber physical systems.

Prof. Petrioli is member of the steering committee of IEEE SECON, was program co-chair of IEEE INFOCOM 2016, workshop co-chair of ACM MobiCom 2014, and general chair of ACM SenSys 2013. She has been member of the steering committee and associate editor of IEEE Transactions on Mobile Computing, member of the steering committee of ACM SenSys, associate editor of IEEE Transactions on Vehicular Technology, member of the executive committee of ACM SIGMOBILE, associate editor of Elsevier Computer Communications, guest editor of special issues for IEEE Access, Elsevier Ad Hoc Networks, Elsevier Physical Communications. She has been program co-chair of leading conferences including ACM MobiCom and IEEE SECON. Prof. Petrioli has published over a hundred fifty papers in prominent international journals and conferences (with over 4850 citations—h-index 37). She has also been PI of over twenty national and international research projects, serving as coordinator of two EC projects (FP7 projects GENESI and SUNRISE). Prof. Petrioli was a Fulbright scholar.

Stefano Basagni is an associate professor at the Department of Electrical and Computer Engineering at Northeastern University, in Boston, MA. He holds a Ph.D. in electrical engineering from the University of Texas at Dallas (December 2001) and a Ph.D. in computer science from the University of Milano, Italy (May 1998). Dr. Basagni's current research interests concern research and implementation aspects of mobile networks and wireless communications systems, wireless sensor networking for IoT (underwater and terrestrial), definition and performance evaluation of network protocols and theoretical and practical aspects of distributed algorithms. Dr. Basagni has published over nine dozens of highly cited, refereed technical papers and book chapters. His h-index is currently 35 (May 2017). He is also co-editor of three books. Dr. Basagni served as a guest editor of multiple international ACM/IEEE, Wiley and Elsevier journals. He has been the TPC co-chair of international conferences. He is a distinguished scientist of the ACM, a senior member of the IEEE, and a member of CUR (Council for Undergraduate Education).

Cynthia A. Phillips received a Ph.D. in computer science from MIT in 1990. She received a B.A. in applied mathematics from Harvard University in 1983. She is currently a Senior Scientist in the Center for Computing Research at Sandia National Laboratories. Dr. Phillips current research areas include combinatorial optimization, algorithm design and analysis, parallel computation, graph algorithms, and streaming algorithms. Dr. Phillips has published over 72 refereed technical journal and competitive conference papers. She has served as a program committee member, sometimes chair, of many international program committees including the International Workshop on Foundations of Mobile Computing (2000-2008, co-chair 2004). She is the vice president for programs for SIAM, is a member of the IEEE TCPP advisory committee and was an officer for the ACM Symposium on Parallelism in Algorithms and Architecture 2000-2004. She received an R&D 100 award in 2006 and in 2008 was a member of a finalist team for the Edeman Award for excellence in operations research practice. She is an ACM distinguished scientist and a SIAM fellow.

Ladislau Bölöni is a Professor of Computer Science at the University of Central Florida. He holds a Ph.D. and M.S. in Computer Science from Purdue University, and B.S. in Computer Engineering from the Technical University of Cluj-Napoca, Romania. He is a Senior Member of IEEE, member of ACM and AAAI. His research interests include artificial intelligence, machine learning, robotics and wireless networking.

Damla Turgut is an Associate Professor at the Department of Computer Science of University of Central Florida. She received her BS, MS, and Ph.D. degrees from the Computer Science and Engineering Department of University of Texas at Arlington. Her research interests include wireless ad hoc, sensor, underwater and vehicular networks, cloud computing as well as considerations of privacy in the Internet of Things. She is also interested in applying big data techniques for improving STEM education for women and minorities. Her recent honors and awards include University Excellence Award in Professional Service in April 2017 and being featured in the UCF Women Making History series in March 2015. She was co-recipient of the Best Paper Award at the IEEE ICC 2013. Dr. Turgut serves as a member of the editorial board and of the technical program committee of ACM and IEEE journals and international conferences. She is a member of IEEE, ACM, and the Upsilon Pi Epsilon honorary society.