Network Optimization for Lightweight Stochastic Scheduling in Underwater Sensor Networks

Dimitri Marinakis, Kui Wu, Ning Ye, and Sue Whitesides

***Abstract*—In this paper, we examine the merit of a simple and lightweight stochastic transmission strategy based on the ALOHA protocol for underwater wireless sensor networks (UWSNs). We use a stochastic scheduling approach in which time is slotted, and each network component transmits according to some probability during each slot. We present objective functions for assigning the transmission probabilities that are aimed at optimizing network performance with respect to the overall network latency and the overall network reliability. We show that there is an easily distributed heuristic policy based on local network density that works well in practice. We also evaluate our approach using numerical simulations. The evaluation results show that even without using explicit control signaling, our lightweight stochastic scheduling method is effective for data transmission in underwater sensor networks.**

***Index Terms*—Underwater senor networks, slotted ALOHA, network optimization.**

I. Introduction

THE monitoring and exploration of the ocean is of great importance to the sustainable and environmentally sound development of the Earth. Activities such as oceanographic data collection, offshore exploration, and ocean ecosystem monitoring are facilitated by the deployment of underwater wireless sensor networks (UWSNs) [9]. One basic requirement is that the devices should be able to exchange data and control messages with each other, as seen in some examples of underwater research platforms (*e.g.*, [2]).

Underwater communication, however, is a challenging problem which is an area of active research (*e.g.*, [14]). Radio waves propagate underwater only at very low frequencies (e.g., 30- 300 HZ) and have an extremely short range (e.g. 20 meters). Under water, optical waves are affected by scattering effects and cannot be used to transmit over long distances. So far, acoustic communication has been the physical layer of choice for underwater communication. Underwater acoustic communication, however, is subject to large propagation latency, low bandwidth, high bit error rate (BER), and complex multipath fading. These special properties call for solutions that are significantly different from those designed for traditional RF communications. To make the situation worse, there can be large variations in temperature, salinity, and pressure over short distances in the underwater environment, all of which can significantly impact acoustic propagation.

In existing RF communication systems, Medium Access Control (MAC) protocols are used to resolve contention issues in medium access. A basic requirement for a MAC protocol is to find a transmission scheduling scheme that eliminates or minimizes conflicting transmissions. To achieve this, an implicit control mechanism (*e.g.*, Time Division Multiple Access (TDMA)), or explicit control messages (*e.g.*, Request-To-Send (RTS) and Clear-To-Send (CTS) messages in Carrier Sense Multiple Access (CSMA) based protocols), are adopted. These scheduling methods are designed, however, for RF communications in the air and usually assume that the propagation delay is negligible. Simulation studies have shown that the RTS/CTS based control, which alleviates hidden/exposed terminal problems and improves network throughput [7], actually degrades throughput when the propagation delay becomes large [24].

Interestingly, it has recently been shown that MAC protocols based on the relatively simple ALOHA protocol [1] perform well in an underwater multi-hop environment in which there are significant propagation delays. Recently, Syed *et al.* [21] modified the slotted ALOHA protocol for underwater, acoustic communication, so that ALOHA could achieve a throughput comparable to what it achieves in RF networks. In related work, Petrioli *et al.* [13] evaluated various MAC protocols for underwater sensor networks and found that in multi-hop, underwater acoustic networks, ALOHA variants performed better than protocols that have larger overhead costs. Additionally, simulations reported by Zhou *et al.* [24] demonstrated that random ALOHA schemes can provide stable performance in UWSNs.

Motivated by the important role we believe ALOHA based MAC protocols will play in UWSNs, we have examined the merit of a simple, stochastic transmission strategy based on the ALOHA protocol: time is slotted; and during each time slot, each device in the network is assigned a probability for transmission. Such a simple link scheduling method is easy to implement and requires virtually zero control overhead. Therefore, we propose that stochastic variants of slotted ALOHA could be used for networking devices in UWSNs.

Main Contributions:

- We lay the groundwork for exploring whether stochastic scheduling for lightweight, ALOHA based MAC variants might provide suitable solutions for underwater network

Manuscript received April 20, 2011; revised November 9, 2011 and March 5, 2012; accepted April 24, 2012. The associate editor coordinating the review of this paper and approving it for publication was S. Shakkottai.

This is a much extended version of our conference paper published in IFIP Networking 2011.

D. Marinakis, K. Wu (corresponding author: wkui@ieee.org), and S. Whitesides are with the Department of Computer Science, University of Victoria, Canada.

N. Ye is with the Department of Information Science, Nanjing College for Population Programme Management, China. This work was done when she was a visiting scholar in the Department of Computer Science, University of Victoria.

Digital Object Identifier 10.1109/TWC.2012.052412.110740

communication challenges. Specifically, we present a Lightweight Stochastic Scheduling (LiSS) approach.

- We consider how the transmission probability of a node should be adjusted based on its local (at a given time) communication topology in order to obtain good overall network performance.
- We present heuristic objective functions for assigning the transmission probabilities to improve overall network performance. We present analytical results and perform simulation study to demonstrate the effectiveness of our Lightweight Stochastic Scheduling.

The remainder of this paper is organized as follows. We review related work on underwater MAC protocols. Then we present the analysis that motivates an easily distributed heuristic for assigning transmission probabilities, and we present experimental results based on network simulations. We conclude with several observations and remarks.

II. Related Work

A number of MAC protocols have been proposed to handle the special conditions encountered in underwater multi-hop sensor networks (*e.g.*, [11], [20], [12], [15]). Here we review the more recent efforts that are particularly relevant to our investigation.

A. Underwater MAC protocols

Despite the disadvantages of the inherent propagation delays, a number of modern MACs proposed for underwater communication nevertheless rely on the exchange of handshaking control messages for medium access. For example, Slotted Floor Acquisition Multiple Access (Slotted FAMA), as presented by Molins and Stojanovic [11], is based on carrier sensing. Each network component constantly listens to the channel, but stays idle unless it has permission to transmit, which is granted via an RTS/CTS handshaking mechanism. Collisions are handled through a random back off scheme. Simulations demonstrate that this protocol has promise for underwater networks, although the authors consider an application in which the data packets exchanged are much larger than the control packets used for the handshaking. It is not clear that this approach would be suitable for an application in which many small data packets are exchanged on a frequent basis.

The Tone-Lohi (T-Lohi) MAC introduced by Syed *et al.* [20] also employs a synchronized transmission frame with a handshaking scheme for contention avoidance. Unlike Slotted FAMA, however, the protocol allows network components to sleep for energy saving purposes. When a device using the T-Lohi protocol is ready to send data, it attempts to reserve the channel by sending a control message (a tone) during a reservation period. If the device does not hear one or more tones from other devices during this reservation period then it is clear to send; otherwise it backs off and waits. Energy savings through sleeping are achieved by using custom acoustic hardware which triggers the device to wake up when the tone is detected.

Considerable research has demonstrated the promise of underwater MAC layers that incorporate CDMA (*e.g.*, the work of Pompili *et al.* [15], the work of Page and Stojanovic [6], and the work of Tan and Seah [22]). The approach is particularly suited for some challenging application areas, such as shallow water operation where multi-path interference is a major factor. In other applications, however, *e.g.*, where congestion issues dominate, the operational simplicity of ALOHA schemes can be attractive.

In contrast to CDMA based approaches, Slotted FAMA and T-Lohi, the Underwater Wireless Acoustic Network Media Access Control (UWAN-MAC) protocol presented by Park and Rodoplu [12] does not employ a handshaking mechanism using control messages to reserve channel access. When using UWAN-MAC, each device transmits infrequently, but regularly with a randomly selected offset. The schedule of a device's neighbor is learned via synchronization packets sent during an initialization period. The approach achieves energy savings by finding locally synchronized schedules such that network components can sleep during idle periods. Although there is no explicit method for avoiding collisions, the collisions are shown to be rare. The approach relies, however, on a static network in which the transmission delays between any pair of devices remain roughly constant. The UWAN-MAC approach has some similarity in spirit to the stochastic scheduling we consider in this paper, and it should be possible to modify UWAN-MAC to benefit from our analysis, *e.g.*, by adapting the duty cycle of each device based on local network density.

B. Stochastic Scheduling

In previous work [10], Marinakis and Whitesides used a slotted stochastic transmission strategy in the context of an alarm network. They addressed the question of how a network of devices might signal the occurrence of an event capable of disabling the sensors. The approach was for the devices to regularly exchange messages during normal operation, but signal the occurrence of an alarm event by ceasing all transmissions. It is concluded that good performance could be obtained by using a heuristic that sets the probability of a node i transmitting per slot to the inverse of the max degree of itself and all its neighbors plus one:

$$p_i = \frac{1}{\max \delta(k) + 1}, k \in \{N(i), i\}, \tag{1}$$

where $N(i)$ denotes the neighboring nodes of node i and $\delta(k)$ means the degree of node k. This heuristic is called the *Max Neighbourhood Degree Heuristic (MNDH)*. The objective of this heuristic is to limit the transmission rate of each device to that of the most overloaded device in its neighbourhood.

The probabilistic approach we present in this paper has some relation to stochastic techniques applied to RF sensor networks such as flooding (see *e.g.*, Sasson *et al.* [16]), or data aggregation (see *e.g.*, Boyd *et al.* [4]). Also related are distributed, low complexity approaches to scheduling such as the recent work of Tang *et al.* [23]. Nevertheless, our work differs from the above in that we consider different objective functions. Furthermore, we obtain analytical results, for which simple heuristics can be designed for distributed scheduling.

III. Network Model

We model the multi-channel, multi-hop communication links available between the network nodes at any instance as a directed graph $G = (V, E)$ in which each vertex $v \in V$ represents a network node and each edge $e_{ij} \in E$ denotes a *potential* communication link from node i to node j, *i.e.*, the node i can transmit data to node j if $e_{ij} \in E$ in a selected channel.

We make the following assumptions on data communication:

1) A device may select either to transmit on a channel m, or to tune its acoustic transceiver to receive data on a channel m during a particular slot. A device that is transmitting may not receive at the same time.
2) If a device is tuned to receive on a channel m, then a packet can be received only if exactly one of its neighbors is transmitting on that channel. This constraint provides a simple way to model congestion issues such as the hidden terminal problem [3].
3) All devices maintain synchronized clocks and may select to time their communications to occur during a particular slot. Note that this is a common assumption, and there are a number of existing techniques that could be used to accomplish this task; see Syed and Heidemann [19] for an example of a time synchronization method appropriate for acoustic networks, and see Sivrikaya and Yener [17] for a more general survey of time synchronization techniques in wireless sensor networks. Note that time synchronization is a basic requirement for a networked system and correct time is required to make sensing data meaningful.
4) Time is slotted, and at each time slot, a device may select to transmit on a randomly selected channel with a given probability, which will be determined according to different performance goals. When not transmitting, a node tunes itself to receive on a randomly selected channel. Note that we do not assume any control channel to coordinate nodes' transmission, because it has been demonstrated before that introducing control mechanisms does not necessarily improve throughput when the propagation delay is large [24].

Since no explicit control mechanism is required, we will refer to this approach as *Lightweight Stochastic Scheduling (LiSS)*.

IV. LiSS on A Single Channel

In this section, we analyze the single channel case. As opposed to solving the congestion issue through the assignment of deterministic schedules, we propose assigning a probability of transmitting per time slot to each node in the network, *i.e.*, $P = \{p_i, i \in V\}$. The key question is how to select appropriate values for P. We will show that our approach leads to good heuristics that can be used to design simple and distributed scheduling that do not depend on the propagation delay. For ease of reference, we list the main notations in Table I.

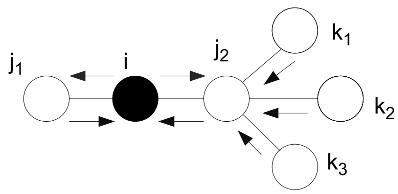


Fig. 1. Example graph showing the influence of p_i. If p_i is adjusted upwards, then the throughput across links (i, j_1) and (i, j_2) will be increased, while that of the links (j_1, i), (j_2, i), (k_1, j_2), (k_2, j_2) and (k_3, j_2) will be decreased.

A. Basic Constraint and General Guideline

As a preliminary, we consider the impact of P on the probability of one node communicating with another. To this end, we define a throughput graph corresponding to a given network.

Definition 1: The **throughput graph** of a given network $G = (V, E)$ is a weighted, directed graph, denoted by $G' = (V, E, R)$, where R is the set of weights on the edges and the weight of edge $e_{ij} \in E$, denoted by r_{ij}, corresponds to the probability that node j receives a message from a neighbouring node i during a given time slot. Although we specify a directed graph for notational convenience, we assume that the edge e_{ij} exists if and only if the edge e_{ji} exists. A node j is called a neighboring node of a node i if there is an edge $e_{ji} \in E$.

We call G' the throughput graph because the weight assigned to the directed edge e_{ij} is proportional to the amount of data across that link in a long run. In this paper, we consider only *static* assignments of values to P. As illustrated in Appendix, r_{ij} can be calculated as follows:

$$r_{ij} = p_i(1 - p_j) \prod_{k \in N(j), k \neq i} (1 - p_k), \quad e_{ij} \in E. \tag{2}$$

The above formula captures the basic constraint. The impact of this constraint is illustrated in Figure 1. In particular, while a node's transmitting probability is increased, the long-term throughput from this node to its neighboring nodes will be increased, and such an increase may lead to lower transmission opportunities for its neighboring nodes.

B. Different Objectives for Value Assignment of P

We now consider how to assign P values for two useful objective functions that aim to reduce link level packet loss and link level latency. For the moment, our objective function formulations weigh equally the importance of all links in the network. Considering a specific subset of network links, for example, a spanning tree, would be a straightforward adaption.

Definition 2: The **overall reliability** of network $G = (V, E)$, denoted by Q_r, is a function over the corresponding throughput graph $G' = (V, E, R)$ and is defined as:

$$Q_r = \prod_{r_{ij} \in R} r_{ij}. \tag{3}$$

We call Q_r the overall network reliability because in the long run, maximizing Q_r should maximize the probability that a data packet could be successfully routed along an arbitrary path q in $|q|$ time slots, where $|q|$ is the length of the path.

TABLE I
MAIN NOTATIONS

Symbols	Meaning
$N(i)$	the neighbors of device i
r_{ij}	the weight of edge e_{ij} corresponding to the probability that node j receives a message from node i in a time slot
p_i	the probability that node i transmits in a time slot
P	the vector of p_i's
δ_i	the degree of node i
Q_r	the overall reliability defined by (3)
Q_l	the overall latency defined by (4)
M	the number of channels
β	a parameter controlling the convergent rate towards the direction of gradient
α	the communication ratio determining the density of a network

Definition 3: The **overall latency** of network $G = (V, E)$, denoted by Q_l, is a function over the corresponding throughput graph $G' = (V, E, R)$ and is defined as:

$$Q_l = \sum_{r_{ij} \in R} \frac{1}{r_{ij}}. \tag{4}$$

Intuitively, $\frac{1}{r_{ij}}$ represents the long-term average delay associated with communication from node i to node j, because the value r_{ij} gives the probability of successfully transmitting a packet from node i to node j in a given time slot. By minimizing Q_l, we obtain value assignment of P that minimizes the delay associated with successfully routing a data packet along an arbitrary path in the long term. The overall network latency is a reasonable objective function in order to design good heuristics to achieve small average routing delay in the whole network. This is particularly useful when the source and destination nodes are not given *a-priori*. We would like to emphasize that Q_r and Q_l are both *heuristic* functions useful to achieve good performance in the long term for a network with unknown traffic patterns.

C. Maximizing Overall Network Reliability

We will now show how to assign the values in $P = \{p_i, i \in V\}$, such that the overall network reliability Q_r is maximized. Since Q_r is a function of P, we rewrite it as $Q_r(P)$ to make the point clear:

$$Q_r(P) = \prod_{r_{ij} \in R} r_{ij} \quad .$$

By taking the log of both sides we get:

$$Q'_r(P) = \sum_{r_{ij} \in R} \log r_{ij},$$

where $Q'_r(P) = \log Q_r(P)$. Note that for $Q_r(P) > 0$, which is for all nontrivial $Q_r(P)$, logarithm is defined. Moreover, the logarithm function is a monotonically increasing function; hence the same $maximizer(P)$ is achieved. It is easy to see that the function Q'_r is a concave function, since if we expand $\log r_{ij}$, each term in the function is concave and the sum of concave functions is also concave.

Now we would like to find the values of P that maximize Q'_r. We proceed by considering the partial derivatives of Q'_r with respect to the value of $p_i \in P$:

$$\nabla Q'_r = \left(\frac{\partial Q'_r}{\partial p_1}, \frac{\partial Q'_r}{\partial p_2}, \ldots \frac{\partial Q'_r}{\partial p_n} \right)$$

where $n = |V|$. A single partial then becomes:

$$\frac{\partial Q'_r}{\partial p_i} = \sum_{r_{lj} \in R} \frac{1}{r_{lj}} \frac{\partial r_{lj}}{\partial p_i} \quad . \tag{5}$$

The partials with respect to p_i are only non-zero, however, for $r_{ij}, j \in N(i)$ and $r_{ji}, j \in N(i)$ and $r_{kj}, k \in N(j), k \neq i$, based on the basic constraint as defined in (2). We can now consider the partial of a single weight value with respect to p_i. The first type of term in (5) with a non-zero partial derivative that we need to consider is the outbound throughput from i to j:

$$\begin{aligned} \frac{\partial r_{ij}}{\partial p_i} &= \frac{\partial}{\partial p_i} p_i (1 - p_j) \prod_{k \in N(j), k \neq i} (1 - p_k) \\ &= (1 - p_j) \prod_{k \in N(j), k \neq i} (1 - p_k) \\ &= \frac{r_{ij}}{p_i} \quad . \end{aligned} \tag{6}$$

For the inbound throughput from j to i where $j \in N(i)$ we have:

$$\begin{aligned} \frac{\partial r_{ji}}{\partial p_i} &= \frac{\partial}{\partial p_i} (1 - p_i) p_j \prod_{k \in N(i), k \neq j} (1 - p_k) \\ &= -p_j \prod_{k \in N(i), k \neq j} (1 - p_k) \\ &= \frac{-r_{ji}}{1 - p_i} \end{aligned} \tag{7}$$

and similarly, for the throughput from k to j where $k \in N(j), k \neq i$ we have:

$$\begin{aligned} \frac{\partial r_{kj}}{\partial p_i} &= \frac{\partial}{\partial p_i} p_k (1 - p_i)(1 - p_j) \prod_{l \in N(j), l \neq i, l \neq k} (1 - p_l) \\ &= -p_k (1 - p_j) \prod_{l \in N(j), l \neq i, l \neq k} (1 - p_l) \\ &= \frac{-r_{kj}}{1 - p_i} \quad . \end{aligned} \tag{8}$$

Let us denote the set of throughputs whose partial derivative with respect to p_i is not trivially zero as L_i. This set can be described as all throughputs that have an endpoint adjacent to either the vertex i or one of its neighbors:

$$L_i = \{r_{kl}, l \in \{N(i) \cup i\} \quad . \tag{9}$$

We can further categorize the throughputs affected by p_i into those with a positive partial derivative:

$$L_{i+} = \{r_{ik}, k \in N(i)\} \tag{10}$$

and those with a negative partial derivative:

$$L_{i-} \quad = \quad L_i \setminus L_{i+} \quad . \tag{11}$$

To further aid our analysis, we adopt δ_i to be the degree of node $i \in V$ and further define $\bar{\delta}_i$ to be the sum of the degrees of all neighbors of i:

$$\bar{\delta}_i = \sum_{j \in N(i)} \delta_j \quad .$$

It is easy to see that $|L_{i+}| = \delta_i$ and $|L_{i-}| = \bar{\delta}_i$.

Let us now return to (5) and consider taking the partial derivative of Q'_r with respect to p_i:

$$\begin{aligned} \frac{\partial Q'_r}{\partial p_i} &= \sum_{r_{jk} \in L} \frac{\partial r_{jk}}{\partial p_i} \frac{1}{r_{jk}} \\ &= \sum_{r_{jk} \in L_{i+}} \frac{\partial r_{jk}}{\partial p_i} \frac{1}{r_{jk}} + \sum_{r_{jk} \in L_{i-}} \frac{\partial r_{jk}}{\partial p_i} \frac{1}{r_{jk}} \\ &= \sum_{r_{jk} \in L_{i+}} \frac{r_{jk}}{p_i} \frac{1}{r_{jk}} + \sum_{r_{jk} \in L_{i-}} \frac{-r_{jk}}{1-p_i} \frac{1}{r_{jk}} \\ &= \frac{\delta_i}{p_i} - \frac{\bar{\delta}_i}{1-p_i} \quad . \end{aligned} \tag{12}$$

Since the partial derivative of our objective function Q'_r with respect to a single p_i does not depend on the other elements of P, we can set each partial to zero in order to find the value of each p_i that will maximize Q'_r and of course Q_r as well:

$$p_i = \frac{\delta_i}{\delta_i + \sum_{j \in N(i)} \delta_j} \quad . \tag{13}$$

The result is a feasible solution, *i.e.*, $0 \leq p_i \leq 1$.

D. Minimizing Overall Network Latency

To minimize overall network latency, we need to solve the following non-trivial optimization problem:

$$\begin{aligned} \underset{p_1, p_2, \ldots, p_n}{\text{minimize}} \quad & Q_l = \sum_{r_{ij} \in R} \frac{1}{r_{ij}} \\ \text{subject to} \quad & 0 < p_i \leq 1, \; i = 1, \ldots, n \end{aligned} \tag{14}$$

where r_{ij} is defined by (2). For each node i, if we introduce another parameter $q_i = 1 - p_i$, we can transform the above problem to an equivalent optimization problem as follows.

$$\begin{aligned} \underset{p_1, \ldots, p_n, q_1, \ldots, q_n}{\text{minimize}} \quad & Q_l = \sum_{r_{ij} \in R} \frac{1}{r_{ij}} \\ \text{subject to} \quad & 0 < p_i \leq 1, \; i = 1, \ldots, n \\ & 0 < q_i \leq 1, \; i = 1, \ldots, n \\ & p_i + q_i = 1, \; i = 1, \ldots, n \end{aligned} \tag{15}$$

Note that after the transformation, the term $(1 - p_i)$ in Q_l, $i = 1, \ldots, n$, should be replaced with q_i. This transformation of the objective function allows us to use geometric programming [5] techniques to solve the problem. Unfortunately, the above optimization problem is not immediately solvable with geometric programming because the equality constraints $p_i + q_i = 1, (i = 1, \ldots, n)$ is not monomial [5]. To avoid this problem, we relax the optimization problem by replacing the equality constraints $p_i + q_i = 1, (i = 1, \ldots, n)$ with inequality constraints $p_i + q_i \leq 1, (i = 1, \ldots, n)$. We can then use geometric programming to find the optimum of the relaxed optimization problem [5].

We next show that the global optimum to the relaxed problem is also the global optimum to the original problem as stated in (15).

Proposition 1: The global optimum to the relaxed optimization problem obtained by replacing the equality constraints $p_i + q_i = 1, (i = 1, \ldots, n)$ with inequality constraints $p_i + q_i \leq 1, (i = 1, \ldots, n)$ in (15) is also the global optimum to the original problem in (15).

Proof. We prove by contradiction. Denote

$$S = (p_1, q_1, p_2, q_2, \ldots, p_n, q_n)$$

as the global optimum to the relaxed problem. If there is an i such that $p_i + q_i < 1$, then we can replace q_i with $q'_i = 1 - p_i$. For each item $\frac{1}{r_{xy}}$ in Q_l, where $x, y = 1, 2, \ldots, n$, if it does not include q_i, its value is unchanged. Otherwise, its value will be decreased since $q'_i > q_i$. Therefore, the value of the objective function Q_l calculated using

$$\tilde{S} = (p_1, q_1, p_2, q_2, \ldots, p_i, q'_i, \ldots, p_n, q_n)$$

will decrease. Since $\tilde{S}$ also meets all the constraints in the relaxed problem, S cannot be the global optimum. This causes a contradiction and as such $p_i + q_i = 1$ must hold for all $i = 1, \ldots, n$. In other words, the global optimum to the relaxed problem is found only under the boundary conditions of $p_i + q_i = 1$.

E. An Iterative Gradient-Following Algorithm

Although geometric programming can be used to obtain the optimal P values for minimizing the overall network latency, the solution is not in closed-form and can not easily be distributed. In the following, we show that there is a clear gradient that can be identified and followed in order to find locally optimal values for the function.

In a manner similar to the calculations carried out in the optimization of overall network reliability, we can arrive at the following expression for a single partial derivative:

$$\frac{\partial Q_l}{\partial p_i} = \sum_{r_{ij} \in R} \frac{\partial r_{ij}}{\partial p_i} \frac{-1}{(r_{ij})^2} \quad . \tag{16}$$

As for (5) in the previous section, the partials with respect to p_i in (16) are only non-zero for $r_{ij}, j \in N(i)$ and $r_{ji}, j \in N(i)$ and $r_{kj}, k \in N(j), k \neq i$. Using (6), (7), (8), (10), and (11), we can write:

$$\begin{aligned} \frac{\partial Q_l}{\partial p_i} &= \sum_{r_{ij} \in L_{i+}} \frac{-1}{p_i r_{ij}} + \sum_{r_{ij} \in L_{i-}} \frac{1}{(1-p_i) r_{ij}} \qquad (17) \\ &= \frac{-1}{p_i} \sum_{r_{ij} \in L_{i+}} \frac{1}{r_{ij}} + \frac{1}{1-p_i} \sum_{r_{ij} \in L_{i-}} \frac{1}{r_{ij}} \quad . \end{aligned}$$

After some algebra it can be seen that the partial derivative is zero when:

$$p'_i = \frac{\sum_{r_{ij} \in L_{i+}} \frac{1}{r_{ij}}}{\sum_{r_{ij} \in L_{i-}} \frac{1}{r_{ij}} + \sum_{r_{ij} \in L_{i+}} \frac{1}{r_{ij}}} \quad . \tag{18}$$

A fixed point of (18) translates to a zero point of the gradient of the utility function $Q_r'(P)$

Given an arbitrary feasible assignment of P values, one can calculate the L values and then identify a gradient that moves towards the target P values:

$$\nabla = (p_1' - p_1, p_2' - p_2, \ldots, p_{|V|}' - p_{|V|}) \tag{19}$$

which we can follow to determine a local minima for the function Q_l. This approach of formulating the gradient takes advantage of the fact that we can find a closed form solution for the zeros of the partials (18). The following stable iterative gradient-following algorithm can now be used to find suitable P values for the network:

P_0 = initial values based on (13).
for t = 1: numOfIterations
1. Calculate each value in R as a function of P_{t-1}.
2. Calculate the target P' as a function of R using (18).
3. Calculate ∇ using (19).
4. Set $P_t = P_{t-1} + \beta(\nabla)$.
end

The parameter β in Step 4 determines the rate at which the algorithm moves in the direction of the gradient. As a general guideline, a smaller β value requires a longer running time (*i.e.*, more iterations) for the algorithm to approach local optimum, but a large β may, on the other hand, make the optimization algorithm never reach the optimum. We will refer to P values obtained using the above algorithm as *gradient-optimized*.

Comparing (18) to (13), it can be seen that the optimal values for each objective function should be similar when the link throughput values across the network are of roughly equal value. Given the formulation of the objective function, it would seem that the variance across the Q_l optimized throughput values varies locally as a function of network density, with the higher variance occuring in the more sparse areas of the network. As a result, the solution to Q_r makes an excellent starting point in the above gradient-based search algorithm. By constructing a gradient in this manner and seeding the above approach with initial values obtained using maximizer values for Q_r (i.e., (13)), we consistently obtained the same results as the global optimal with geometric programming in less than 20 iteration steps with a reasonable β value of 0.1. In addition, for multiple problem instances, where we select numerous arbitrary initial values, as well as initial value using (18), all trials for the same problem instance converge to the same mode.

F. A Heuristic to Approximate Gradient-Optimized Values

We propose a closed-form approximation algorithm based on two facts: (1) The above gradient-based search algorithm consistently obtains the same result as the global optimum obtained using geometric programming; (2) using the solution to maximizing Q_r as the initial values quickly converges to an optimal solution in the gradient search algorithm. Therefore, we use the values that optimize Q_r given in (13) as a heuristic to approximate optimal values for the overall network latency.

If we divide both the numerator and denominator of (13) by δ_i, we get the inverse of one plus the average degree of i's neighbors, *i.e.*, node i should transmit at each time slot with a probability equal to

$$p_i = \frac{1}{1 + \frac{\sum_{j \in N(i)} \delta_j}{\delta_i}} \tag{20}$$

where $N(i)$ represents the neighbors of device i, and δ_i gives the degree of node i. We refer to this assignment of P values as the *Average Neighbourbood Degree Heuristic* (ANDH). This is as opposed to the Max Neighbourhood Degree Heuristic (MNDH) suggested in the prior work [10] which considered the maximum degree of all of a node's neighbors, *i.e.*, node i should transmit at each time slot with a probability equal to

$$p_i = \frac{1}{1 + \max_{j \in \{N(i), i\}} \delta_j}. \tag{21}$$

We will show later in numerical simulations, that the ANDH gives a close approximation to gradient-optimized values obtained using the iterative approach given in the previous section.

V. LiSS on Multiple Channels

In the case where we have $M (M > 1)$ frequency channels, we assume that each network device tunes its transceiver to one of the M channels selected uniformly at random each time step. In this case the value for R in G' will take on different values. We can calculate the value of r_{ij} as follows:

$$\begin{aligned} r_{ij} &= \binom{M}{1} \frac{1}{M} p_i \frac{1}{M} (1 - p_j) \prod_{k \in N(j), k \neq i} (1 - \frac{1}{M} p_k), e_{ij} \in E \\ &= p_i \frac{1}{M} (1 - p_j) \prod_{k \in N(j), k \neq i} (1 - \frac{1}{M} p_k), e_{ij} \in E \end{aligned} \tag{22}$$

where $N(x)$ returns the neighbors of x according to G. This is because in order for node i to successfully transmit a packet to node j over a particular channel, nodes i and j need to tune to that channel for transmitting and receiving, respectively, and node j's neighboring nodes (except node i) should not transmit over that channel. The probability that the first condition holds is $\frac{1}{M} p_i \frac{1}{M} (1 - p_j)$, and the probability that the second condition holds is $\prod_{k \in N(j), k \neq i} (1 - \frac{1}{M} p_k), e_{ij} \in E$. The $\frac{1}{M}$ term specifies the probability of the transceiver being tuned to that particular channel. The term $\binom{M}{1}$ means there are M ways to select a particular channel among M channels.

Following the same analysis steps as in the single channel case, we can obtain the P values that maximize overall network reliability or minimize overall network latency. Interestingly, the maximizing P values for overall network reliability are the same as those found in the single channel case.

Similarly as in the single channel case, we can use geometric programming to obtain the global optimum solution to minimize the overall network latency. We can also obtain equations for the gradient which can be used to find (locally) optimal values of P to minimize overall network latency.

Although it makes the analysis tractable, the use of a uniform distribution for channel selection is not necessarily the best choice. For example, in a dense region of the network in which each device has many neighbors. For the moment, we leave further investigation into arbitrary, device specific channel distributions as future research.

VI. Approximated Distributed Algorithms for LiSS

The analytical results in the previous section provide us with heuristics to design simple, distributed algorithms for LiSS. In this section, we present two distributed algorithms that are aimed at maximizing overall network reliability and minimizing overall network latency, respectively.

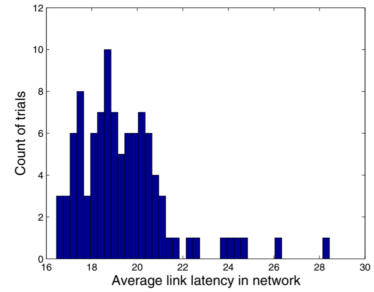
A. An Approximated Distributed Algorithm to Maximize Overall Network Reliability

It is straightforward to assign P values in a distributed manner in order to optimizing overall network reliability. (13) and (22) depend only on the knowledge of the local communication topology and the number of frequency channels employed. We present the following distributed algorithm for selecting a suitable value of the transmission probability of an individual device.

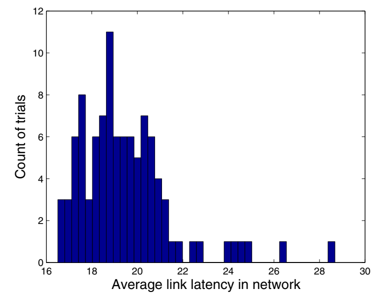
1) During initial deployment, a default value for p_i can be assigned to each node given a rough estimate of the typical network density.
2) Each node maintains a neighbour table with one entry for each of its neighbors. Each neighbour table entry includes the unique media access control (MAC) identification of the neighbour, along with the timestamp and the number of neighbors reported by that neighbour.
3) Each node exchanges neighbour count estimates with each of its neighbors, and then updates its transmission probability p_i accordingly.
4) At each time slot, with probability p_i, on a channel selected uniformly at random, a node broadcasts its unique media access control (MAC) identification, the number of entries in its neighbour table, and any data payload.
5) If not transmitting, the node tunes its receiver to a channel selected uniformly at random, and if it receives a message, it adds the appropriate details to its neighbour table and updates it's p_i values accordingly.

B. An Approximated Distributed Algorithm to Minimize Overall Network Latency

The distributed algorithm adopts the gradient-based approach for selecting P values that locally optimize overall network latency. In order to update its p_i value, each node requires knowledge of its two-hop communication topology and both the transmission probability values and throughput values of its two-hop neighbors. Given this information a node can calculate its local gradient ∇ and adjust its p_i value in the direction of the gradient according to a suitable rate $\beta : p_i = p_i + \beta\nabla$, where β can be adjusted to change the speed of moving in the direction of the gradient.



(a)



(b)

Fig. 2. Distribution of average link delay in units of time slot duration: (a) Gradient-Optimized, (b) ANDH. Data from 100 trials of 100-node networks constructed with a communication ratio $\alpha = 0.25$, which results in an average node degree of 5.49.

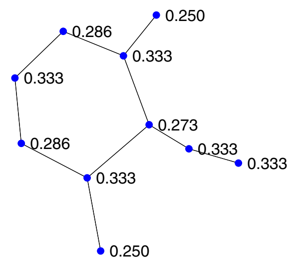
VII. Performance Evaluation

We perform simulation studies to evaluate LiSS. In the simulation, we build the network topology using *disk graphs*. The graphs are obtained by selecting points uniformly at random in a region of the plane bounded by a circle of diameter D as the locations of the network nodes. An edge is then assigned between any two vertices if the pair-wise distance between their associated locations is less than a given ratio α of the deployment diameter D. Since the parameter α can be used to control the number of communication links of the network, we call it *communication ratio*. We assign a delay to an edge proportional to the distance between the pairs (but rounded to a discrete value for ease of simulation). These types of graphs are commonly used as models in sensor network research (see *e.g.*, Gandham et al. [8]). To ease presentation, we only show performance results over a single channel.

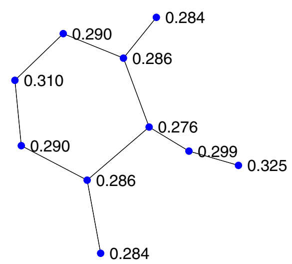
Since the gradient-based search algorithm always obtained the same results as the global optimum with geometric programming in all our simulation trials, we in the following only show the gradient-optimized values to make the figures easy to read.

A. Assignment of P values: Gradient-Optimized vs. Approximate Gradient-Optimized

As shown in Figure 2, the P values obtained using the ANDH were close to those obtained using the Gradient-Optimized values, suggesting that the ANDH is a good approximation for optimizing values of the overall network latency. The expected delay across each link was found by computing the per link throughput according to (2) as a function of the network topology. Figure 3 shows an example of the P values obtained using these two techniques for a small



(a)



(b)

Fig. 3. A small example network topology with P values assigned using: (a) ANDH and (b) Gradient-Optimized.

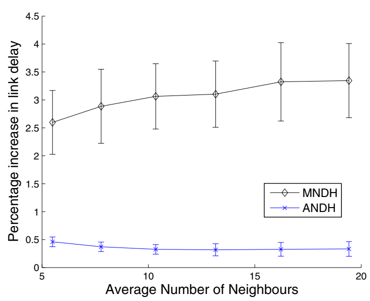


Fig. 4. Average percentage increase in link delay values over Gradient-Optimized P values using ANDH and MNDH. Results were averaged over 100 trials of 100-node networks for each communication ratio considered. Error bars depict one standard deviation.

example communication topology. In addition, as shown in Figure 4, for networks of different node densities, the ANDH consistently results in lower average latency than that obtained with the MNDH, which was introduced in [10].

As an additional note, for each of the trials considered in this set of experiments, we ran the gradient based approach for finding the locally optimal P values to minimize overall network latency twice, one using random initial values, and one using values seeded by the ANDH. In all cases the two runs converged to the same result.

B. Performance under varying Load and Propagation Delay

For this test, we control the traffic load by assigning each node a probability that the node has a data packet given to its MAC layer at the beginning of each time slot. We selected T-Lohi [20] for comparison because this protocol represents a typical example of using lightweight control packets (*i.e.*, it uses a RTS control packet only). Our implementation of the T-Lohi algorithm is as described in 'Algorithm 1' of [20] using a single time slot as the duration of a contention round.

As shown in Figure 5(a), T-Lohi performs better than LiSS under light traffic load, because T-Lohi employs control messages for media access. Nevertheless, LiSS performs much better under heavy traffic load. This demonstrates that using control packets may not be ideal for some underwater acoustic communication scenarios, since the propagation delay is determined by the pair-wise distance between the two communicating nodes, whose locations may be uncertain. The resulting randomness in delay can make the effective use of control packets challenging.

In Figure 5(a), in order to investigate the average transmission delay, we assumed a very large queue size to avoid buffer overflow. It can be seen that the average transmission delay of LiSS with the ANDH is quite stable under various traffic load. The average transmission delay with T-Lohi, however, increases drastically under heavy traffic load. To further study how many packets could be delivered within a given time constraint, we changed the queue size to one. In other words, if the queue already contains a pending packet then any additional packets to be sent are dropped. Figure 5(b) shows the average percentage of a node's neighbors that receive a packet sent by that node, under different traffic loads. We can also see that the performance of LiSS with the ANDH outperforms T-Lohi when traffic load becomes heavy.

The results shown in Figures 5(a) and (b) are obtained by setting the propagation delay much smaller than the length of a time slot. We then increase the propagation delay considerably to further investigate its impact. In particular, we set the max propagation delay in the network to three times the length of a time slot. Under these conditions, we simulate T-Lohi sending RTS control packets, but do not assume that the protocol knows in advance the per link propagation delay. Figure 5(c) shows the results under this set of tests. Comparing Figure 5(b) and Figure 5(c), it can be seen that LiSS is almost un-affected, but the T-Lohi protocol suffers considerably. This is due to the fact that the RTS control packets used by the T-Lohi protocol may fail to reserve the channel due to propagation latency. In this set of experiments the average transmission delay and the percentage of packets delivered were empirically measured using our simulation framework.

To summarize, the stochastic nature of our approach makes it well suited for applications where propagation delay is long and random. For such applications, when traffic load becomes heavy, protocols that rely on control packets to resolve medium contention may not perform well.

VIII. Conclusions and Future Work

In this paper we present and evaluate the concept of using a lightweight variant of slotted ALOHA in conjunction with a stochastic scheduling approach, called LiSS, in which network nodes transmit according to some probability each slot. We consider how to assign the transmission probabilities in order to minimize the overall network latency or maximize the overall network reliability. We obtain a closed-form solution

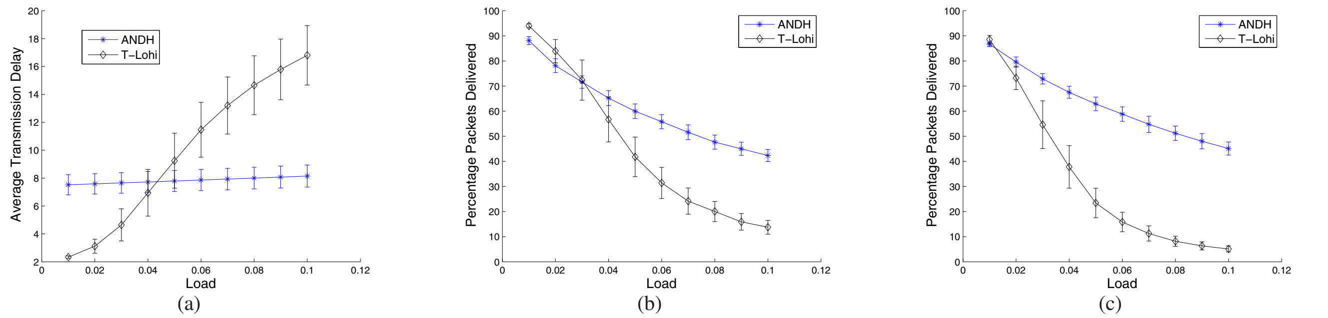


Fig. 5. (a) Average time from queuing a packet until its delivery (T-Lohi *vs.* LiSS using the ANDH); (b) Percentage of successful packet transmissions over *all* links as a function of traffic load; and (c) the same as as (b) but under the condition of long propagation delay. Results were averaged over 100 trials of 100-node networks using a communication ratio $\alpha = 0.25$. Error bars depict one standard deviation.

for the maximization of overall network reliability and show that geometric programming can be used to minimize the overall network latency. We also present distributed algorithms that can be used to find suitable transmission probabilities. Performance results demonstrate that even without using any control signaling, LiSS works well for UWSNs where propagation delay is not negligible.

Modeling channel fading of underwater acoustic communications is another interesting research challenge, and a cross-layer optimization considering channel fading and dynamic channel states will be our research work.

Acknowledgment

We would like to thank Prof. Wusheng Lu in the Department of Electrical and Electronic, University of Victoria for introducing geometric programming, and the reviewers for their valuable comments.

Appendix

Using a detailed example, we explain why (2) could be used to calculate r_{ij}, oblivious to the propagation delay.

We assume that the propagation delay between two nodes does not change over time if their distance is fixed. Let us assume that the propagation delay between two nodes, i and j, is a discrete number of time slots and is specified by d_{ij}. This is a good approximation because the time slot is practically much smaller than the propagation delay for underwater acoustic communication. For example, transmission speed of an underwater transceiver could be 40 kbps [18]. Assume that the maximum frame size is 1 kb. If we set the time slot as the maximum time for transmitting one frame, a time slot is about 25 ms. Depending on the depth, the salinity, and other factors, propagation speed of acoustic wave under water may vary, but we may assume a commonly-acceptable value of 1500 m/s. For two wireless nodes with the distance of 10 km (a reasonable value for underwater sensor networks such as that in the Neptune project [2]), the propagation delay is 6.6 seconds, much larger than the time slot 25 ms.

Consider node i in Fig. 1 as an example. Three conditions must hold such that at time slot t node i receives a frame from node j_1: (1) at time slot $t - d_{j_1 i}$ node j_1 transmits the frame, (2) at time slot t, node i does not transmit and receive the frame, and (3) at time slot $t - d_{j_2 i}$ node j_2 does not transmit

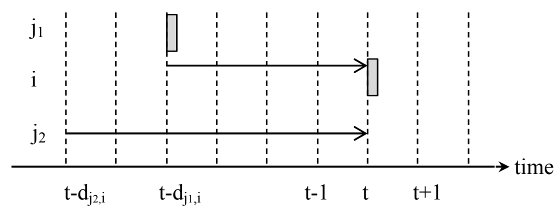


Fig. 6. Time relationship illustrating the conditions that node i receives a message from node j_1.

since otherwise two frames would collide at node i at time slot t. The time relationship is illustrated in Fig. 6.

Therefore, if we assign p_l^m to be the transmission probability of node l at time slot m, the probability that at time slot t node i receives a frame from node j_1, denoted as $r_{j_1 i}^t$, can be calculuated as

$$r_{j_1 i}^t = p_{j_1}^{t-d_{j_1 i}}(1 - p_i^t)(1 - p_{j_2}^{t-d_{j_2 i}}), \tag{23}$$

where the terms $p_{j_1}^{t-d_{j_1 i}}$, $(1-p_i^t)$, and $(1-p_{j_2}^{t-d_{j_2 i}})$ correspond to the first, the second, and the third conditions, respectively. Since we assume a static assignment to P values, e.g., $p_{j_1}^{t-d_{j_1 i}} = p_{j_1}$ for all time slots. The above equation can be rewritten as:

$$r_{j_1 i} = p_{j_1}(1 - p_i)(1 - p_{j_2}). \tag{24}$$

That is, the propagation delay does not influence the calculation of $r_{j_1 i}$.

References

[1] N. Abramson, "Development of the alohanet," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 119–123, Mar. 1985.

[2] C. R. Barnes, J. R. Delaney, B. M. Howe, and N. Penrose, "Neptune: a regional cabled observatory in the northeast pacific," white paper for Ocean Research Interactive Observatory Networks (ORION) meeting, Jan. 2004.

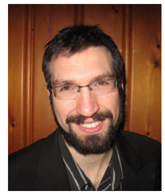
[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "Macaw: a media access protocol for wireless LANs," in *Proc. 1994 ACM SIGCOMM*, pp. 212–225.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory and IEEE/ACM Trans. Networking*, vol. 52, no. 6, pp. 2508–2530, June 2006.

[5] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and Engineering*, vol. 8, no. 1, pp. 67–127, 2007.

[6] E. C. Page and M. Stojanovic, "Efficient channel-estimation-based multi-user detection for underwater acoustic CDMA systems," *IEEE J. Oceanic Engineer.*, vol. 33, no. 4, pp. 502–512, Oct. 2008.

[7] C. L. Fullmer and J. Garcia-Luna-Aceves, "Solutions to hidden terminal problems in wireless networks," in *Proc. 1997 ACM SIGCOMM*, pp. 39–49.
[8] S. Gandham, E. Dawande, and R. Prakash, "Link scheduling in sensor networks: distributed edge coloring revisited," in *Proc. 2005 IEEE INFOCOM*, vol. 4, pp. 2492–2501.
[9] L. Liu, S. Zhou, and J. Cui, "Prospects and problems of wireless communications for underwater sensor networks," *Wireless Commun. and Mobile Computing*, vol. 8, no. 8, pp. 977–994, Oct. 2008.
[10] D. Marinakis and S. Whitesides, "The sentinel problem for a multi-hop sensor network," in *Proc. 2010 Canadian Conference on Computer and Robot Vision*, pp. 253–260.
[11] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC protocol for underwater acoustic networks," in *Proc. 2006 IEEE Oceans Conference*, pp. 1–7.
[12] M. K. Park and V. Rodoplu, "UWAN-MAC: an energy-efficient MAC protocol for underwater acoustic wireless sensor networks," *IEEE J. Oceanic Engineering*, vol. 32, no. 3, pp. 710–721, Jan. 2007.
[13] C. Petrioli, R.Petroccia, and M.Stojanovic, "A comparative performance evaluation of MAC protocols for underwater sensor networks," in *Proc. 2008 IEEE Oceans Conference*, pp. 1–10.
[14] D. Pompili and I. F. Akyildiz, "Overview of networking protocols for underwater wireless communications," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 97–102, Jan. 2009.
[15] D. Pompili, T. Melodia, and I. Akyildiz, "A distributed CDMA medium access control for underwater acoustic sensor networks," in *2007 Med-Hoc-Net*.
[16] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. 2003 WCNC*, vol. 2, pp. 1124–1130.
[17] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July-Aug. 2004.
[18] M. Stojanovic, "Recent advances in high-speed underwater acoustic communications," *IEEE J. Oceanic Engineer.*, vol. 21, no. 2, pp. 125–136, 1996.
[19] A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. 2006 IEEE INFOCOM*, pp. 1–2.
[20] A. Syed, W. Ye, and J. Heidemann, "T-Lohi: a new class of MAC protocols for underwater acoustic sensor networks," in *Proc. 2008 IEEE INFOCOM*, pp. 231–237.
[21] A. Syed, W. Ye, B. Krishnamachari, and J. Heidemann, "Understanding spatio-temporal uncertainty in medium access with ALOHA protocols," in *Proc. 2007 WUWNet*, pp. 41–48.
[22] H.-X. Tan and W. K. Seah, "Distributed CDMA-based MAC protocol for underwater sensor networks," in *Proc. 2007 IEEE Conf. on Local Computer Networks*, pp. 26–36.
[23] S.-J. Tang, X. Wu, X. Mao, Y. Wu, P. Xu, G. Chen, and X.-Y. Li, "Low complexity stable link scheduling for maximizing throughput in wireless networks," in *Proc. 2009 SECON*, pp. 1–9.
[24] Z. Zhou, Z. Peng, and J. Cui, "Multi-channel MAC protocols for underwater acoustic sensor networks," in *2008 International Workshop on UnderWater Networks(WUWNet)*.



Dimitri Marinakis received the Ph.D. degree from the McGill University, Montreal, Quebec, Canada. He was a post-doc fellow in the Department of Computer Science, University of Victoria, Canada during 2010-2011 and is currently a researcher at Kinsol Research Inc. His research interests include applying statistical and probabilistic methods to problems in sensor networks.

Kui Wu received the Ph.D. degree in Computing Science from the University of Alberta, Canada, in 2002. He joined the Department of Computer Science at the University of Victoria, Canada in the same year and is currently an Associate Professor there. His research interests include mobile and wireless networks, network performance evaluation, and network security. He is a senior member of the IEEE and a member of the ACM.

Ning Ye received the BS degree in Computer Science from NanJing University, China, in 1994, the MS degree in School of Computer & Engineering from Southeast University, China, in 2004, and the Ph.D. degree in Institute of Computer Science from Nanjing University of Post and Telecommunications, China, in 2009. She worked as a Visiting Scholar and Research Assistant in the Department of Computer Science, University of Victoria, Canada, in 2010. She is now an Associate Professor in the Department of Information Science of Nanjing College for Population Programme Management, China and holds an adjunct position at Nanjing University of Post and Telecommunications, China. Her research focuses on wireless networks and pervasive computing. She is a senior member of Chinese Computer Federation (CCF).

Sue Whitesides received an MS in Electrical Engineering from Stanford University in 1969 and a Ph.D. in Mathematics from the University of Wisconsin-Madison in 1975. She has served as a faculty member of the mathematics departments of Tufts University and Dartmouth College and of the School of Computer Science of McGill University. She is currently Professor of Computer Science at the University of Victoria, Canada. Her research interests include computational geometry, design and analysis of algorithms, graph drawing, motion planning, combinatorial mathematics, and applications of these areas to a variety of other fields. She is a member of the ACM, AMS, and SIAM.