

Towards Optimal Broadcasting Policies for HARQ based on Fountain Codes in Underwater Networks

Paolo Casari, Michele Rossi and Michele Zorzi

Department of Information Engineering, University of Padova, 6/B Via G. Gradenigo, I-35131 Padova (PD), Italy

E-mail: {casari,rossi,zorzi}@dei.unipd.it

Invited paper

Abstract—This paper explores hybrid ARQ policies based on Fountain Codes for the transmission of multicast messages in underwater channels. These rateless codes are considered because of two nice properties, namely, they are computationally lightweight and do not require to know the channel erasure probabilities at the receivers prior to transmission. In this paper, these codes are used together with a Stop and Wait ARQ to enhance the performance of broadcast communications. First, we present a dynamic programming model for the characterization of optimal broadcasting policies. The derived broadcasting rules are then compared against plain ARQ schemes via Monte-Carlo simulation.

Our results show that digital fountain codes are a promising technique for the transmission over underwater channels as their performance, in terms of delay, reliability and energy efficiency, clearly dominates that of plain ARQ solutions. This paper is a preliminary study on the topic and encourages us towards the design of practical HARQ protocols for the underwater medium.

Index Terms—Underwater acoustic networks, broadcast, fountain codes, hybrid ARQ, optimal transmission policy.

I. INTRODUCTION

Underwater acoustic sensor networks (UWASNs) are becoming increasingly popular as a research area in telecommunications due to the many constraints that a transmission protocol has to face in this environment. Specifically, sufficiently long communication ranges and reliability are currently possible through the use of acoustic waves. However, the propagation of sound underwater incurs extremely long delays and only allows limited bit rates. These facts need to be taken into account in the design of any communication scheme, especially when the network operates according to the *ad hoc* paradigm.

Research on networking protocols for UWASNs can be considered in its infancy. Previous work was done regarding MAC protocols [1]–[3] and some preliminary studies [4], [5] provide solutions for routing. There is, however, room for further work in terms of analysis and protocol design. In fact, the underwater channel has peculiar characteristics which entail the re-design or, in some cases, the invention of completely new solutions. Moreover, fundamental tradeoffs involving the use of transmission power, available bandwidth within channel access as well as the use of retransmissions and forward error

correction (FEC) at the link layer are often unexplored by the related literature. Along these lines, in this paper we look into techniques for reliable broadcasting in underwater networks. This is a fundamental service which received little attention to date, as the only available study to the best of our knowledge consists of the practical schemes in [6].

In this work we apply a new paradigm to underwater broadcasting, incorporating effective coding techniques such as the recently proposed fountain codes [7] into Automatic Repeat reQuest (ARQ) error recovery. Our main objective is to limit, as much as possible, the number of transmissions to reliably disseminate a message to a number of nodes. This is particularly critical in UWASNs, due to the high cost of transmissions [8]. In this paper, our hybrid ARQ (HARQ) solution is first modeled using dynamic programming [9]. This analysis is subsequently used to obtain optimal retransmission policies under maximum delay constraints. Optimal HARQ is finally compared with plain ARQ to measure the performance improvements which are achievable under the assumption of ideal channel access. Nevertheless, our analysis as well as the obtained results are general and can be used, in future research, for the design of practical broadcasting schemes.

The paper is structured as follows. Section II presents a review of the related work in the field of network protocol design for UWASNs. In Section III we describe the underwater channel and review a model for the calculation of packet error probabilities according to channel and transmission parameters. Section IV introduces the fountain codes we use in this paper. In Section V we detail our mathematical framework, through which we obtain optimal error recovery policies for fountain-based HARQ for broadcasting in underwater channels. In Section VI simulation results are shown to demonstrate the effectiveness of these HARQ schemes, their superiority with respect to plain ARQ, as well as relevant tradeoffs. Finally, Section VII concludes the paper.

II. RELATED WORK

The use of acoustics for underwater communication has received increased interest in recent years. While the main use of acoustic waves is still sonar detection and ranging, as well as telemetry [10], relatively recent efforts have proven that reliable

This work is supported in part by NOAA's Sea Grant College Program, Project no. NA060AR4170019.

links can be set up in water, using signal processing techniques that provide good communication efficiency or speed [11]–[14].

There are still many open issues in building underwater acoustic networks [15]. Most of the research work done so far focused on the design of MAC protocols. A discussion of deterministic multiple access schemes for underwater networks was presented in [16]. A more comprehensive comparison of such schemes in clustered environments has been more recently carried out in [17]. Other protocols have been more specifically tailored to the underwater acoustic channel features. For example, Slotted FAMA [18] focuses on collision avoidance. It sets up shared synchronization among the sensors, whereby the time is divided into slots sufficiently long to accommodate for the maximum round-trip time in the network. Transmissions are preceded by an RTS/CTS handshake, and may take place only at the beginning of a slot. A number of protocols followed [1]–[3], [19]. PCAP focuses on collision avoidance by making the duration of handshakes predictable; this allows the transmitter to carry out other tasks while waiting for the receiver to reply. In [1] collision control is sought instead of avoidance, mainly through the exchange of signaling messages prior to data transmission. The protocol in [1] cannot avoid collisions completely. However, the reduced length of the waiting times ensures a globally greater throughput, outperforming Slotted FAMA. Moreover, there is no need to maintain node synchronization. UWAN-MAC [2] is designed to save energy through very low duty cycles, and focuses on collision avoidance through some sort of adaptive TDMA. Time synchronization with neighbors is achieved through the transmission of special packets. HELLO packets are used to recover from synchronization errors, such as waking up and hearing no transmission by the intended sender. The authors in [20] argue that the difference between transmit and receive power can be exploited in underwater networks and discuss how to manage idle time in light of this. The conclusion is that near-optimal energy performance can be reached if ultra-low power transducer wakeup modes could be implemented. On a similar line of thought, Tone-Lohi [3] tries to avoid collisions by sending very short busy tones, that could be heard by other nodes during idle channel monitoring. Optimal packet sizes for ARQ protocols as well as different variants of Stop and Wait ARQ (*i.e.*, operating over single vs. multiple channels) were investigated in [21]. The tradeoffs discussed in that paper are key to the design of any reliable network protocol in UWANs.

Research on more complex network protocols for, *e.g.*, routing or broadcasting is very recent. We describe below the few research efforts in this sense. We observe that most of the literature is focused on the adaptation of terrestrial radio protocols to the underwater environment. Segmented Data Reliable Transport (SDRT) [4] employs FEC to guarantee error protection. Each node encodes and forwards data continuously using a simplified version of Tornado codes, until some positive feedback is received. To avoid wasting too much energy, packet transmissions are “windowed:” the packets inside the window are transmitted at full rate, whereas a lower rate is used for those outside the window. Each receiver must decode the whole block of data before transmitting again. In [5],

the authors deploy a framework for addressing delay-sensitive and -insensitive applications, involving Reed-Solomon packet coding and scheduling of packets according to their delay requirements. The focus of the investigation is on the impact of the long delays and stronger attenuation of the acoustic channel on packet routing. The variation of the available bandwidth with distance is taken into account in [22], where the authors find an optimal transmission distance in terms of energy consumption for line and three-dimensional topologies.

In this paper we continue the line of research on networking protocols for UWANs by studying HARQ schemes for broadcasting in underwater channels. To the best of our knowledge the only previous paper dealing with this problem is [6]. The most advanced protocol in that paper has Forward Error Correction capability, as we assume here. However, the scope and the approach of [6] are complementary to those of the present paper. First, rather than devising a practical protocol, as the authors do in [6], we focus on the theoretical gains that are achievable through the use of coding and ARQ. Moreover, we model in detail an important class of codes (namely, fountain codes) that were not considered in [6]. The work we present here highlights tradeoffs and potential gains which can be exploited in the proposal of new practical schemes.

Differently from [5], we use fountain codes as they allow for a lightweight implementation of encoder and decoder. In fact, low computational complexity is key for unmanned, battery operated underwater sensors. We also observe that fountain codes are rateless, which means that the optimal amount of redundancy to generate and send for recovery can be decided through online algorithms. This is very important, especially given the high communication cost, in terms of power and delay, in underwater acoustic channels.

III. CHANNEL MODEL

The physics of acoustic propagation must be taken into account for a proper design of underwater communication networks. Acoustic waves have a peculiar propagation behavior [23]. First of all, they propagate at a slow speed $c \approx 1500$ m/s, which is five orders of magnitude smaller than radio propagation in the air. The propagation speed actually changes with the depth, temperature, and salinity of the water, but we will consider it fixed for simplicity.

The most unusual feature of the underwater acoustic channel is the dependence between bandwidth and transmission distance [24]. More precisely, the available bandwidth shrinks for increasing distances due to a superposition of frequency-dependent effects related to attenuation and noise. We explain this starting with Urlick’s model for the attenuation incurred by a tone at frequency f as a function of the distance d [23]:

$$A(d, f) = d^k a(f)^d. \quad (1)$$

k is the spreading coefficient, and models the geometry of the propagation. If the propagation is perfectly spherical (such as in deep water where a wave finds no boundaries until after several kilometers), $k = 2$. Conversely, if the propagation is perfectly cylindrical (such as in very shallow water), $k = 1$. Typically,

$k = 1.5$ is chosen to represent a mixed propagation scenario. It is observed that k is the counterpart of the attenuation exponent in the radio path loss model. Finally, the factor $a(f)$ in (1) is the absorption loss, and models the conversion of acoustic pressure into heat due to the resonance with certain ions present in the water. This factor can be approximated by Thorp's formula [25]:

$$\mathcal{A}(f) = \frac{0.11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + 2.75 \cdot 10^{-4}f^2 + 0.003, \quad (2)$$

where $\mathcal{A}(f) = 10 \log_{10} a(f)$. Equation (2) returns $a(f)$ in dB/km for f in kHz. From the above equations, we observe that the attenuation increases with frequency and that the dependence on distance is much stronger than in radio channels, due to the exponential term $a(f)d$ in (1).

The noise power spectral density (psd) depends on the frequency as well, being composed of four main contributions: turbulence (subscript t), shipping (s) and other human activities, wind and waves (w), and thermal noise in the receiver circuitry (th). They can be modeled as follows:

$$\begin{aligned} \mathcal{N}_t(f) &= 17 - 30 \log(f) \\ \mathcal{N}_s(f) &= 40 + 20(s - 0.5) + 26 \log(f) - 60 \log(f + 0.03) \\ \mathcal{N}_w(f) &= 50 + 7.5\sqrt{w} + 20 \log(f) - 40 \log(f + 0.4) \\ \mathcal{N}_{th}(f) &= -15 + 20 \log(f), \end{aligned} \quad (3)$$

where $\mathcal{N}_x(f)$ stays for $10 \log_{10} N_x(f)$, whereas s is the *shipping factor*, representing the intensity of shipping activities on the surface of the water, and has values ranging between 0 and 1. The factor w is the wind speed in m/s. The total noise psd is $N(f) = \mathcal{N}_t(f) + \mathcal{N}_s(f) + \mathcal{N}_w(f) + \mathcal{N}_{th}(f)$. The different components impact the noise psd at different frequencies.

We are now ready to define the average SNR of a tone transmitted at a frequency f and traveling a distance d as [24]

$$SNR(d, f) = \frac{P_T/A(d, f)}{N(f)\Delta f}, \quad (4)$$

where P_T is the transmit power and $N(f)$ is the noise power spectral density (assumed constant in a narrow band Δf around f). In (4), the factor $1/A(d, f)N(f)$, is the frequency-dependent term. Since $A(d, f)$ increases with frequency while $N(f)$ decreases at least to a certain point, the product between the two has a maximum for some frequency f_0 . This maximum has minimal combined attenuation and noise effects (*i.e.*, it is the best frequency to use for transmission).

Figure 1 shows a number of concave grey lines that represent the factor $[A(d, f)N(f)]^{-1}$ for varying d from 10 m to 100 km. Each line corresponds to a different distance; some relevant curves are plotted for illustration. The upper and lower bounds of the available bandwidth $B(d)$ are shown by means of two black lines. $B(d)$ is derived around f_0 according to the -3 dB definition, *i.e.*, $B(d) = \{f : SNR(d, f) > SNR(d, f_0)/2\}$. Notably, the available bandwidth shrinks for increasing distance (*i.e.*, by spanning the black lines from top to bottom). This is different from what happens in radio and must be carefully accounted for in the design of network protocols for UWASNs. For example, performing a few long-range hops in a multi-hop

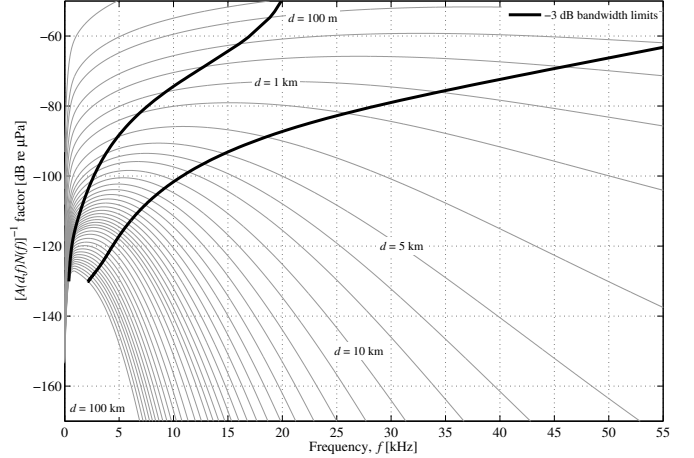


Fig. 1. Frequency-dependent part of the SNR for an acoustic tone transmitted underwater. Two black lines represent the lower and upper limit of the available transmit bandwidth $B(d)$. Grey lines represent the $[A(d, f)N(f)]^{-1}$ factor for different distances.

path would require to transmit at a very high power (because of the large attenuation) and at a low bit rate (because of the small bandwidth), thus increasing the energy consumption considerably.

We now derive the channel effects over a signal with a certain spectrum $S(f)$. Assume for simplicity that this spectrum is flat, *i.e.*, $S(f) = P_T/B(d)$. The SNR in this case is [24]

$$SNR(d, B(d)) = \frac{P}{B(d)} \frac{\int_{B(d)} A^{-1}(d, f) df}{\int_{B(d)} N(f) df}. \quad (5)$$

Due to the one-to-one relationship between d and $B(d)$, all integrals in (5) are determined when d is fixed.

Given the above characterization, let p be the probability of success for a packet containing b bits. According to (5), and assuming the use of a BPSK modulation with independent channel errors on the b transmitted symbols, when the distance between sender and receiver is d we have

$$p = \left(1 - \frac{1}{2} \operatorname{erfc} \sqrt{SNR(d, B(d))}\right)^b. \quad (6)$$

This formula gives the BPSK packet error rate assuming additive white Gaussian noise. Although the noise is non-white [24], see (3), when the signal bandwidth is sufficiently small, one can assume that the power spectral density of the noise is almost constant. In these conditions, expression (6) holds as an approximation. The SNR is defined as that of an *equivalent* AWGN channel as in (5) [24].

IV. INTRODUCTION TO FOUNTAIN CODES

Digital Fountain Codes [7] are high performance codes based on sparse bipartite graphs. These codes are rateless, which means that the amount of redundancy is not fixed prior to transmission but can be rather adjusted on the fly until full recovery. There is no theoretical limit to the number of redundant

packets that can be transmitted. These codes are proven to be asymptotically near-optimal for every erasure channel and very efficient as the size of the message to transmit grows. Also, they operate on packet units by means of simple XOR operations, which allows for extremely efficient implementations. This makes these codes considerably faster than, *e.g.*, classical Reed Solomon codes. Consider a set of K (source) packets, each having the same length of L bits. The encoding procedure works as follows:

- 1) To create an encoded packet t_n , randomly pick a degree d_n from a given degree distribution $\rho(\cdot)$, whose characteristics depend on the set size K , as well as on the targeted performance (*e.g.*, in terms of coding complexity vs. overhead).
- 2) Pick, uniformly at random, d_n distinct input packets from the input message and set t_n equal to the bitwise sum, modulo 2, of these d_n packets. This can be realized by successively XORing the d_n packets. An encoding vector is created for t_n : it contains a list of the original packets that were XORed together. This vector is either transmitted along with t_n or retrieved at the receiving side through intelligent association of the packet identifiers and the random seeds used in the encoding phase (this last method is always preferred in practice).

Decoding can be done by solving the system $\mathbf{t} = \mathbf{G}\mathbf{s}$, in which the matrix \mathbf{G} is formed by the received encoding vectors, the vector \mathbf{t} contains the received encoded packets, and \mathbf{s} contains the K original packets to be retrieved. Recovery at each interested receiver requires the reception of K linearly independent coded packets, so that \mathbf{G} has full rank and can be inverted. As encoded packets are randomly generated according to the degree distribution $\rho(\cdot)$, it is possible that some of the received packets are linearly dependent, so that in general a node needs to receive $K' \geq K$ packets before being able to decode. The performance of these codes, in terms of overhead $O = K' - K$, depends on the degree distribution, and can be kept small by properly designing $\rho(\cdot)$. Reference [26] presents suitable distributions and heuristic decoding methods for large K , that lead to extremely lightweight computations (\mathbf{G} is inverted through heuristic algorithms) as well as a small overhead O . However, for underwater communication, we are constrained to work with small K values. This is in line with the typical message sizes that can be supported in underwater channels (which are small due to the limited bandwidth). In this case, special distributions $\rho(\cdot)$ are to be designed. Here, however, we rather focus on the theoretical gains provided by fountain codes; thus we consider the *digital random fountain* in [7]. Considering a random fountain makes sense here as these codes have asymptotically optimal performance in terms of throughput efficiency. Hence, they allow us to study the maximum improvements which can be achieved through this type of codes in the underwater channel. We note that these codes are not practical in terms of computational complexity, even though practical codes with very good performance in terms of throughput and acceptable computational complexity can be found using optimization approaches similar to that

in [27]. However, the design of practical codes is not within the scope of this paper and is left for future research.

We now define two suitable distributions which will be used in the following analysis. We model the dynamics of the fountain-based encoding/decoding system through a distribution $\psi(x)$ with x integer, returning the probability of correct decoding at a generic user upon the reception of the x -th encoded packet, *i.e.*, full recovery does not occur for packets $x - 1, x - 2, \dots$. Note that $\psi(x) = 0$ for $x < K$ as a full rank (of K) matrix cannot be obtained in this case. We further define $\Psi(x)$ as $\Psi(x) = \sum_{z=0}^x \psi(z)$. We observe that these distributions, given the complexities involved, are usually difficult to obtain in closed-form for actual systems. However, we can empirically measure them. The measured distributions are a simple yet effective method for capturing the complexity of the encoding-decoding process and can be plugged into the analysis that follows. The analysis is thus valid for general fountain codes. Nevertheless, for the random digital fountain that we consider here, from [7] a tight approximation for $\Psi(\cdot)$ is obtained as $\Psi(x) = 1 - 2^{x-K}$ if $x \geq K$ and $\Psi(x) = 0$ otherwise. This specific type of function will be used in the results of Section VI.

V. OPTIMAL RETRANSMISSION POLICIES FOR FOUNTAIN-BASED HARQ

Next, we find the optimal error recovery policy for HARQ systems using fountain codes. We assume to have a message to be broadcast by a transmitting node. This message is subdivided into K (original) packets of fixed length, which are given to the HARQ protocol for their delivery to a number of receivers. Error recovery occurs through a number of transmission rounds. By *optimal error recovery* we mean the use of the optimal number of encoded packets that the transmitting node has to send at each round. Our interest is on optimal online policies; the transmitter, at the generic round i , has to make a decision on the optimal amount of redundancy to send x_i , based on what happened up to and including round $i - 1$.

In the following Section V-A we develop an analysis for the case where all receivers have the same packet error probability p , *i.e.*, they all belong to the same probability *class*. While being of little practical interest, this case provides useful insights on the involved tradeoffs. Moreover, it is the fundamental building block for quasi-optimal policies when users belong to multiple classes. The latter case is analyzed in Section V-B.

A. Optimal Retransmission Policy for Single Class Users

Let R be the number of receiving nodes within the transmission range of the sender. The sender's task is to reliably deliver the original K information packets to these R nodes. This is done through a number of transmission rounds $1, 2, \dots, L$, where L is their maximum number. During the generic round i , the sender transmits a number of encoded packets x_i such that $x_i^{\min} \leq x_i \leq x_i^{\max}$, where $x_1^{\min} = K$, $x_i^{\min} = 1$, for $i > 1$, whereas x_i^{\max} s are design constraints. The number of encoded packets sent to the R neighbors up to and including round i is $X_i = \sum_{j=1}^i x_j$. We refer to $\xi_i(r)$ as the number of packets correctly received, out of these

X_i packets, by the r -th neighbor. The optimal policy, which is the objective of the following analysis, maps X_{i-1} and $m_{i-1} = \min\{\xi_{i-1}(1), \xi_{i-1}(2), \dots, \xi_{i-1}(R)\}$ to the number of additional encoded packets, x_i , we are to send in the current round i in the case where at least one of the neighbors, at the end of round $i-1$, is still not able to decode the original K packets (we call this a *failure* at round $i-1$). Formally, $x_i = \mu_i(m_{i-1}, X_{i-1})$. The goal of our optimization is to find a policy $\mu_i(\cdot)$, for each round i , which minimizes the total number of packets to send so that decoding of the original K packets runs to completion at all neighbors. The cost associated with the transmission process is a function of the round number and x_i , $C(i, x_i)$. Here, we consider $C(i, x_i) = x_i$, which reflects our objective of minimizing the total number of packets needed for full recovery at all neighbors. The controls of our problem are the x_i s, *i.e.*, the number of additional packets sent during round i . Each round i is a decision epoch where, in case of *failure* at round $i-1$, the decision maker (the sender) must pick a number of additional packets for the current round i . This decision is made so as to minimize the expected cost over all possible system dynamics. To carry out this minimization, we define $J(i, m_{i-1}, X_{i-1})$ as the minimum cost-to-go at the beginning of round i , *i.e.*, the cumulative cost incurred from round i until all neighbors can decode, given that the state of the HARQ system *at the beginning* of round i is represented by the pair $\mathcal{S}_i = (m_{i-1}, X_{i-1})$. The exact expression of $J(\cdot)$ is the objective of the following derivations.

Now we introduce some quantities. We assign the same channel erasure probability p to the wireless link connecting the sender to each neighbor $r = 1, 2, \dots, R$. To simplify the notation, we define $\mathcal{B}(x, y, p) = \binom{x}{y} p^y (1-p)^{x-y}$. The joint probability that a given user correctly decodes the K original packets by the end of round $i-1$ and that $\xi_{i-1} = x$, given X_{i-1} , is computed as:

$$\mathcal{P}_s(i-1, x) = \mathcal{B}(X_{i-1}, X_{i-1} - x, p) \Psi(x). \quad (7)$$

Similarly, the joint probability that the generic user r is still unable to decode (unsuccessful decoding, u) at the end of round $i-1$ and that $\xi_{i-1} = x$, referred to as $\mathcal{P}_u(i-1, x)$, is obtained by replacing $\Psi(x)$ in (7) with $1 - \Psi(x)$. At round i , for a given $\mathcal{S}_i = (m_{i-1}, X_{i-1})$ and for a given user r we can derive the probability that $\xi_{i-1}(r) = x$, conditioned on the value m_{i-1} and the *failure* event, f :

$$P(\xi_{i-1}(r) = x | m_{i-1}, f) = \frac{P(\xi_{i-1}(r) = x, m_{i-1}, f)}{P(m_{i-1}, f)}. \quad (8)$$

Due to the symmetry involved, the above equation as well as the following ones do not depend on the specific user r . Hence, for readability, we will omit this dependence in what follows. To compute $P(m_{i-1}, f)$, we first define the joint probability of having a *successful* (s) or *unsuccessful* (u) decoding and that $\xi_{i-1} \geq y$ packets were correctly received at a specific user by the end of round $i-1$, given X_{i-1} , as (accounting for all possible error patterns of length $0, \dots, X_{i-1} - y$ in rounds

$1, 2, \dots, i-1$):

$$F_{i-1, \chi}^{\geq}(y) = \sum_{e=0}^{X_{i-1}-y} \mathcal{P}_{\chi}(i-1, X_{i-1} - e), \quad \chi \in \{s, u\}, \quad (9)$$

Equation (9) is used to find the tail distribution for the minimum m_{i-1} (for brevity denoted as y in the next equation) as follows:

$$\begin{aligned} F_{i-1}^{\geq}(y, R) &= \sum_{r=1}^R \binom{R}{r} F_{i-1, u}^{\geq}(y)^r F_{i-1, s}^{\geq}(y)^{R-r} = \\ &= (F_{i-1, u}^{\geq}(y) + F_{i-1, s}^{\geq}(y))^R - F_{i-1, s}^{\geq}(y)^R. \end{aligned} \quad (10)$$

Note that $F_{i-1, u}^{\geq}(y) + F_{i-1, s}^{\geq}(y) = \sum_{e=0}^{X_{i-1}-y} \mathcal{B}(X_{i-1}, e, p)$ does not depend on $\Psi(\cdot)$. The joint distribution of m_{i-1} and *failure* (f) at the end of round $i-1$ (denominator in (8)) is finally obtained as:

$$P(m_{i-1}, f) = F_{i-1}^{\geq}(m_{i-1}, R) - F_{i-1}^{\geq}(m_{i-1} + 1, R). \quad (11)$$

We now calculate the numerator in (8). To this end, we first compute the tail probability $P(\xi_{i-1} = x, m_{i-1} \geq y, f)$:

$$P(\xi_{i-1} = x, m_{i-1} \geq y, f) = \begin{cases} f(x, y) & 0 \leq y \leq x \text{ and} \\ & 0 \leq x \leq X_{i-1} \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

with:

$$\begin{aligned} f(x, y) &= \mathcal{P}_s(i-1, x) F_{i-1}^{\geq}(y, R-1) + \\ &+ \mathcal{P}_u(i-1, x) \left(\sum_{e=0}^{X_{i-1}-y} \mathcal{B}(X_{i-1}, e, p) \right)^{R-1}. \end{aligned} \quad (13)$$

where the two terms in the previous sum account for full ($\mathcal{P}_s(\cdot)$) or incomplete ($\mathcal{P}_u(\cdot)$) recovery for the user under observation up to and including round $i-1$. Now, $P(\xi_{i-1} = x, m_{i-1} = y, f)$ may be found from (12) as:

$$\begin{aligned} P(\xi_{i-1} = x, m_{i-1} = y, f) &= P(\xi_{i-1} = x, m_{i-1} \geq y, f) - \\ &- P(\xi_{i-1} = x, m_{i-1} \geq y+1, f). \end{aligned} \quad (14)$$

We can now calculate (8), which is subsequently used to define two further equations, $G_{i, u}^{\geq}(x)$ and $G_{i, s}^{\geq}(x)$, representing the joint probability of having a *unsuccessful* (u) or *successful* (s) decoding for a given user by the end of round $i > 1$, and $\xi_i \geq x$, given m_{i-1} :¹

$$\begin{aligned} G_{i, \chi}^{\geq}(x) &= \sum_{z=m_{i-1}}^{X_{i-1}} P(\xi_{i-1} = z | m_{i-1}, f) \left[\sum_{e=0}^{x_i} \mathcal{B}(x_i, e, p) \times \right. \\ &\times \left. g_{\chi}(z + x_i - e) \mathbb{1}\{z + x_i - e \geq x\} \right], \end{aligned} \quad (15)$$

¹For an exact analysis, the packets correctly received by each user up to and including round $i-1$ should be tracked. However, the calculation of the optimal policy is computationally infeasible in this case. To overcome this, we describe the evolution of the process by tracking only the minimum m_{i-1} . We found that the results provided by this approach are very close to those of the exact analysis.

where the total number of packets received up to and including round $i - 1$, X_{i-1} , is bounded by $\sum_{j=1}^{i-1} x_j^{\min} \leq X_{i-1} \leq \sum_{j=1}^{i-1} x_j^{\max}$, the minimum number of correctly received packets, m_{i-1} , is such that $0 \leq m_{i-1} \leq X_{i-1}$, and the number of packets x_i sent in round i must satisfy $x_i^{\min} \leq x_i \leq x_i^{\max}$. Hence, the bounds for x are $m_{i-1} \leq x \leq X_{i-1} + x_i$. Also, $\chi \in \{s, u\}$, $g_s(x) = \Psi(x)$ and $g_u(x) = 1 - \Psi(x)$, $\mathbb{1}\{\cdot\}$ is the indicator function, returning one when the expression within parentheses is true and zero otherwise. It is used here to account only for the cases where $\xi_i \geq x$. The indices z and e track the overall number of packets correctly received in rounds $1, 2, \dots, i - 1$ and the number of erroneous packets out of the x_i sent in round i , respectively. We finally derive the joint distribution of m_i and *failure* at the end of round i , given m_{i-1} (we assume $m_0 = 0$) and that we had a *failure* at round $i - 1$:

$$\Omega(m_i | m_{i-1}) = \begin{cases} P'(m_1, f) & i=1 \\ G_i^{\geq}(m_i, R) - G_i^{\geq}(m_i + 1, R) & i > 1, \end{cases} \quad (16)$$

where $G_i^{\geq}(m_i, R) = (G_{i,u}^{\geq}(m_i) + G_{i,s}^{\geq}(m_i))^R - G_{i,s}^{\geq}(m_i)^R$ (see the derivation in (10)) and $P'(m_1, f)$ is given by (11), by substituting X_{i-1} with x_1 in the related calculations. We are now ready to write the optimality equation (17) for our problem [9]. In (17) $i = 1, 2, \dots, L$, $J(L + 1, \cdot, \cdot) = T$ is the terminal cost we incur in failing to deliver the original K packets in L rounds. Note that the terminal cost should be sufficiently large such that whenever a solution terminating with success (full recovery at all nodes) in strictly less than $L + 1$ rounds exists, this solution will be preferred and consequently found by the dynamic programming optimizer. As before, $\sum_{j=1}^{i-1} x_j^{\min} \leq X_{i-1} \leq \sum_{j=1}^{i-1} x_j^{\max}$, $m_{i-1} = 0, 1, \dots, X_{i-1}$. Moreover, $\ell(i)$ is given by:

$$\ell(i) = \begin{cases} x_1^{\min} & i=1 \\ x_i^{\min} & i > 1 \text{ and } m_{i-1} \geq K \\ \min(K - m_{i-1}, x_i^{\max}) & i > 1 \text{ and } m_{i-1} < K. \end{cases} \quad (18)$$

In fact, in round $i > 1$, if $m_{i-1} < K$ at least $K - m_{i-1}$ additional packets are to be sent so that all receivers r with $\xi_i(r) = m_{i-1}$ can collect at least K packets (which is the minimum number of packets required for decoding). The optimal transmission strategy, *i.e.*, the optimal x_i to use at each round for any state $\mathcal{S}_i = (m_{i-1}, X_{i-1})$, is found by solving backwards (17), see [9].

B. Retransmission Policies for Users belonging to Multiple Probability Classes

Consider now R receivers and C probability classes. Class c has $R_c > 0$ users, all with the same packet error rate p_c and $\sum_c R_c = R$. Let X_{i-1} be the total number of encoded packets transmitted up to and including round $i - 1$ and m_{i-1}^c

be the minimum among the packets correctly received by users of class c by the end of round $i - 1$. For each class c , the optimal online policy $\mu^c(\cdot)$ is found according to the analysis in the previous section. At round i , we choose the number of packets to send as: $x_i = \max_{i=1,2,\dots,C} \{\mu_i^c(m_{i-1}^c, X_{i-1})\}$. This policy, which is very simple and easy to implement in practice, is quasi-optimal. The criterion that we try to satisfy with the above rule is that of satisfying *all* users. In fact, in our case we want to transmit the smallest amount of redundancy that is optimal for the worst case class. The exact calculation of overall optimal policies would involve a straightforward generalization of the analysis in Section V-A. This however leads to a very large state space, which makes the calculation of the optimal policy computationally infeasible.

VI. RESULTS

A. Scenario Description

The fountain codes introduced in Section IV are used along with the optimal retransmission policies of Section V to perform efficient broadcasting in underwater networks. We stress that the high transmission costs and the long propagation delays imposed by the underwater channel constitute a challenge for the efficiency of a broadcast protocol.

To highlight the relevant tradeoffs, we focus here on the following scenario. We consider a source that needs to broadcast a message to R nodes. We assume that these nodes are placed at the same distance from the source, *i.e.*, they all belong to the same probability class.² The broadcast message is composed of $K = 32$ packets of length $b = 1000$ bits, that are to be transmitted within $L = 5$ rounds, according to the analysis in Section V. The minimum and maximum number of packets to send in round $i > 1$ is constant and fixed to $x_i^{\min} = 1$ and $x_i^{\max} = \Delta$, $i > 1$, respectively. At round $i = 1$, $x_i^{\min} = K$ and $x_i^{\max} = K + \Delta$. We set $\Delta = 16$, which provides adequate protection. At the end of each round, each receiver feeds back the overall number of correctly received packets to the source. In this study, we assume that this feedback is error-free: at the end of any round i the transmitter knows exactly which nodes need further transmissions and the value of m_i . The transmission attempts are terminated if some nodes are still unable to decode the broadcast message at the end of round L . In this case we have a transmission failure as the system could not deliver the K original packets to all receivers.

We compare our fountain code-based approach to a plain ARQ protocol working as follows. A maximum of L retransmission rounds is considered. At the first round, the transmitter sends the K original packets and subsequently collects

²It is observed that the protocol performance is largely dominated by the nodes placed farther away from the sender. Hence, considering that all nodes are at the same distance allows to precisely capture the dependence on the distance from the transmitter as well as on the number of receivers R .

$$J(i, m_{i-1}, X_{i-1}) = \min_{\ell(i) \leq x_i \leq x_i^{\max}} \left\{ C(i, x_i) + \sum_{m_i=m_{i-1}}^{m_{i-1}+x_i} \Omega(m_i | m_{i-1}) J(i+1, m_i, X_{i-1} + x_i) \right\}. \quad (17)$$

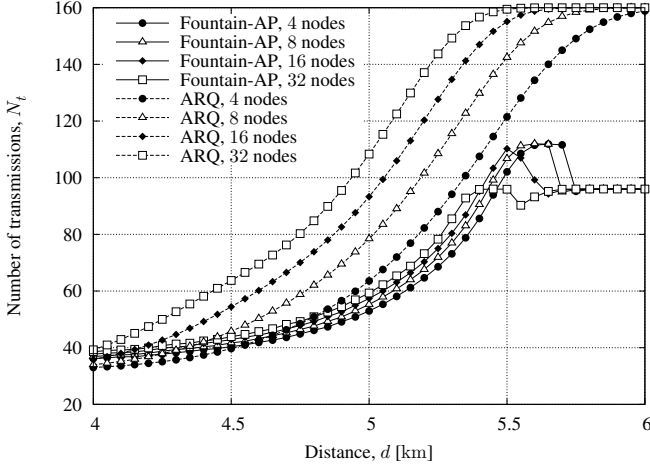


Fig. 2. N_t as a function of d for varying R , AP case.

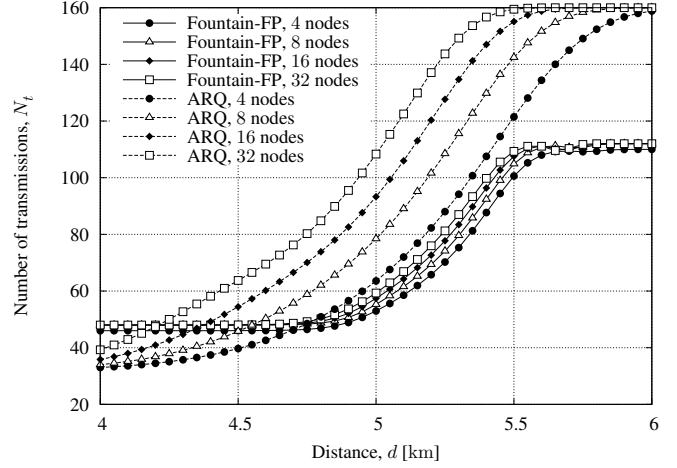


Fig. 3. N_t as a function of d for varying R , FP case.

feedback from the receivers, as explained above. At the next round, only the packets that have been erroneously received or lost by one or more nodes are retransmitted. The performance comparison between the fountain-based ARQ and the plain ARQ schemes is done through Monte-Carlo simulations. For each setting of the involved parameters, we repeated a number of experiments so as to get sufficiently tight confidence intervals about our performance measures. In particular, the 95% confidence intervals for the subsequent plots are all within $\pm 5\%$ of the average values. These intervals are not shown in the graphs for improved readability. The considered metrics are the average total number of packet transmissions N_t , the broadcast failure probability P_{fail} , the average number of rounds N_{rnd} required to complete the delivery process and the corresponding transmission delay D . All metrics are obtained as a function of the distance d , by varying the number of receivers R .

Transmission bandwidth and frequency are chosen according to the channel model in Section III, where the values $s = 0.5$ and $w = 0$ have been used in (3). The transmission rate is 1 kbps. For simplicity we fix the propagation speed of acoustic waves to $c = 1.5$ km/s. The transmission power is set so that the average probability of error per packet at a distance of 5 km from the receiver is 0.25. All results are obtained for a set of distances d from 4 to 6 km, which translates into packet error probabilities ranging from 10^{-4} to roughly 0.9.

B. Simulation Results

We start by looking at the average total number of packet transmissions, N_t . Note that the optimal policies for the fountain code case are obtained so that the total number of transmissions is minimized, under the constraint that the broadcast dissemination is completed within L rounds. This limit on the maximum number of rounds is very important for an underwater environment, where the propagation delay is substantially longer than that experienced over radio channels. Two types of fountain based ARQ are considered in the following. In the first type, an optimal transmission policy is obtained for each distance. This approach is referred to as AP

(for adapted policy) in the figures. Alternatively, one could fix a policy that is optimal for a certain error probability (and thus for a certain distance), and use it regardless of the actual probability value. This policy is referred to as FP (for fixed policy) in the plots. The latter approach may be more desirable in practice, *e.g.*, when nodes have a limited amount of memory and thus cannot store optimal policies for a large set of error probabilities. The results concerning these two cases are shown in Figures 2 and 3, respectively. The performance of plain ARQ is also plotted for comparison. From Figure 2 we see that with an increasing number of receivers R , the fountain approach is much better than plain ARQ. For instance, with 16 users at $d = 5.25$ km, only 75 transmissions are required, compared to ARQ that needs about 128 transmissions. In addition, while the fountain code approach is generally better than plain ARQ, at short distances the fountain code requires more transmissions for full recovery. This is because it has an inherent overhead to allow the reconstruction of the original message, *i.e.*, K' packets have to be correctly received, where K' is slightly larger than K , see Section IV.

This is further confirmed by Figure 3, where the policy for the fountain code is optimal for an error probability of 0.25 (*i.e.*, $d = 5$ km) but is kept fixed for all other distances. Since the error probability at 4 km is much smaller than 0.25, the number of packets sent at the first round is over-dimensioned with respect to the real needs. Due to the excessive overhead sent at low error probabilities (small d) the fountain code is successful (*i.e.*, the matrix G in Section IV has full rank) with high probability. This, however, causes an unnecessary transmission overhead, as can be seen from Figure 3 for $d \leq 4.4$ km.

A second observation is in order here. Recall that encoded packets are obtained by randomly picking a number of original packets and XORing them together. From a practical point of view, this selection is obtained through a random number generator, which is initialized with the same seed at both sender and receivers. A convenient choice of the initial seed is such that the encoding vectors associated with the first K packets sent are linearly independent. In this way, decoding

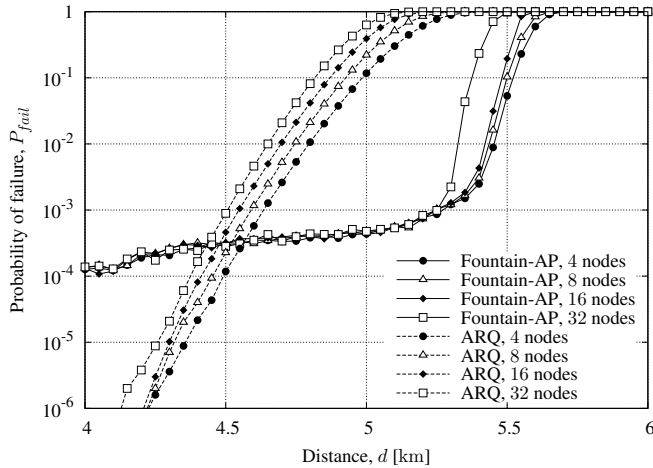


Fig. 4. P_{fail} as a function of d for varying R , AP case.

requires no transmission overhead at short distances. Hence, even though the observed degradation of fountain codes (higher N_t) makes sense from a theoretical point of view, it may not be so important in practice.

Let us now consider the behavior of the fountain code scheme beyond $d = 5.5$ km in the adaptive policy case (Figure 2). There, the number of transmissions reaches its maximum before decreasing to around 96. This is peculiar to the way the policy is derived, and can be better explained with the aid of Figure 4, which shows the average failure probability, P_{fail} . From this graph, we see that the probability that the broadcast is in fact completed within L rounds is still sufficiently high up to 5.5 km. This means that the optimal policy in these cases will always try to send enough packets to allow full recovery. Between 5.75 and 6 km, instead, the per-packet error probability becomes very large and the policy deems it unworthy to send too many packets. In fact, these packets would be lost with high probability, thereby only wasting resources. Under these very high error rates the optimal policy sends the minimal amount of allowed redundancy, *i.e.*, $K - m_{i-1}$ packets, where m_{i-1} is the minimum number of correct packets at round $i - 1$, as dictated by the third line in (18). Observe that in this case the optimal policy would send no packets at all if the lower bound $\ell(i)$ in (18) were not imposed. This bound is however required to obtain proper policies for lower error rates.

Figure 4 is also interesting because of the error probability floor of the fountain approach at low d . This is due to the adopted optimization criterion, which achieves the minimum number of transmissions, not the minimum error probability. It is observed that the policy which minimizes the error probability is very simple and corresponds to transmitting exactly $x_1^{max} = K + \Delta$ packets at the first round and $x_i^{max} = \Delta$ packets at each following round, $i > 1$.

We proceed by analyzing the average number of rounds required to complete the broadcast transmission, N_{rnd} . Figures 5 and 6 show results for the AP and the FP cases, respectively. It is observed that AP is better than plain ARQ. Nevertheless, note that the number of transmission rounds is not minimized

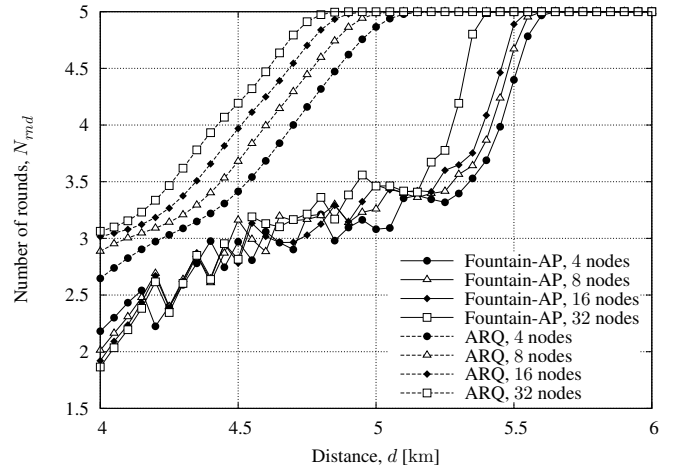


Fig. 5. N_{rnd} as a function of d for varying R , AP case.

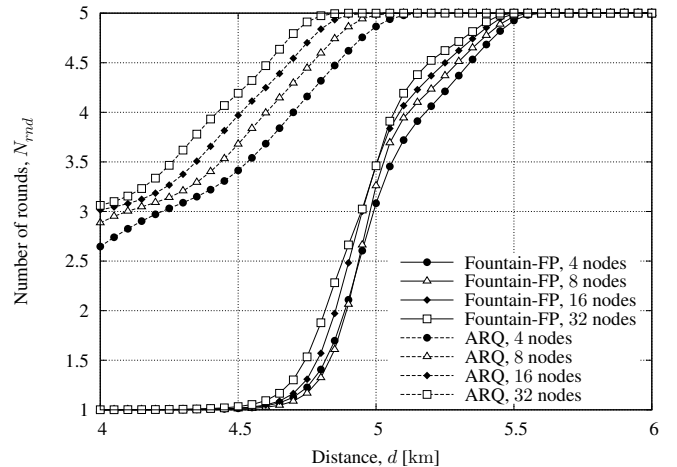


Fig. 6. N_{rnd} as a function of d for varying R , FP case.

by the transmission policy. As for the failure probability, this is due to the optimality criterion used to derive the policy, according to which we minimize the overall number of packets transmitted within L rounds. In detail, in the first round the transmitter sends a limited number of packets and compensates for residual errors if needed. Again, the optimal policy for minimizing N_{rnd} would consist of transmitting the maximum number of packets at the beginning of each round. As we see in Figure 6 at short distances, when the number of transmissions is over-dimensioned the fountain code achieves full recovery in just one round. We finally observe that the oscillations shown by the fountain-based ARQ for $d \leq 5$ km is correct and due to the fact that the optimal behavior, in terms of transmission overhead, consists of using the same policy until the error probability reaches a certain critical value, after which the policy is changed. Discontinuities in N_{rnd} occur at these points. If a fixed policy is used for all distances, as we do in Figure 6, this phenomenon is absent.

We now discuss the average delay to complete the broadcast, D , which is shown in Figure 7 (AP case). This delay is obtained

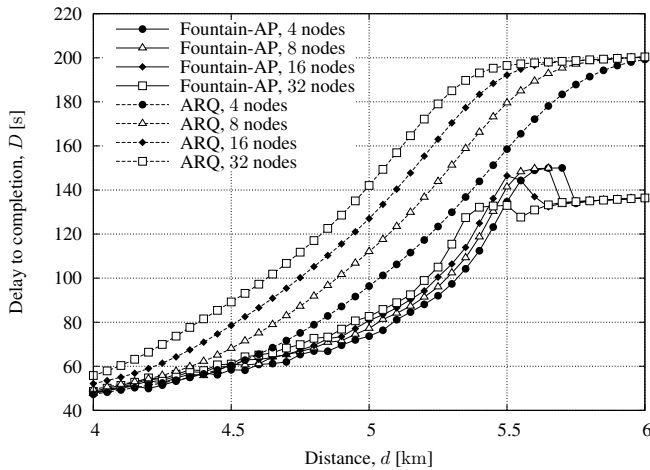


Fig. 7. D as a function of d for varying R , AP case.

considering all sources of delay, including the propagation time. These results look similar in shape to those of Figure 2 and confirm the superiority of HARQ and the importance of using efficient communication techniques for transmission in the underwater environment. Our fountain-based approach proves to be very effective in reducing the broadcast delay in all but a very limited set of scenarios (small d). Practical considerations can be implemented to account for these cases as well. These are, however, left for future research.

We conclude our study observing that HARQ, as expected, scales better for increasing R , while the performance of ARQ is substantially impacted.

VII. CONCLUSIONS

In this paper we presented a preliminary investigation of the theoretical performance improvements offered by hybrid ARQ in underwater communication networks. As the specific coding technique, we focused on fountain codes, a family of rateless codes that is lightweight and easy to use within online retransmission policies. We formulated the transmission process using dynamic programming, obtaining optimal policies which minimize the number of transmissions needed for full recovery at all receivers. We compared our scheme with a plain ARQ policy, concluding that fountain codes present a number of advantages which can be leveraged to suit the peculiar needs of underwater communications.

Future directions of this work include the extension of the presented results to general topologies as well as the design of a practical schemes for the dissemination of broadcast messages in multihop underwater networks.

REFERENCES

[1] B. Peleato and M. Stojanovic, "A MAC protocol for ad hoc underwater acoustic sensor networks," in *Proc. ACM WUWNet*, Los Angeles, CA, USA, Sep. 2006, pp. 113–115.

[2] M. K. Park and V. Rodoplu, "UWAN-MAC: an energy-efficient MAC protocol for underwater acoustic wireless networks," *IEEE J. Ocean. Eng.*, 2007, to appear. [Online]. Available: http://www.ece.ucsb.edu/rodoplu/Pubs/ParkRodoplu_UWANMAC.pdf

[3] A. Syed, W. Ye, and J. Heidemann, "Medium Access Control for Underwater Acoustic Networks," in *Proc. ACM WUWNet*, Los Angeles, CA, USA, Sep. 2006, work-in-progress paper.

[4] P. Xie and J.-H. Cui, "SDRT: a reliable data transport protocol for underwater sensor networks," UCONN CSE, Tech. Rep. UbiNet-TR06-03, Feb. 2006.

[5] D. Pompili, T. Melodia, and I. F. Akyildiz, "Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks," in *Proc. ACM Mobicom*, Los Angeles, CA, Sep. 2006, pp. 298–309.

[6] P. Casari and A. F. Harris III, "Energy-efficient reliable broadcast in underwater acoustic networks," in *ACM WUWNet*, Sep. 2007.

[7] D. J. C. MacKay, "Fountain Codes," *IEE Proceedings – Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[8] L. Freitag and M. Grund and S. Singh and J. Partan and P. Koski and K. Ball, "The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms," <http://www.whoi.edu>, 2005.

[9] S. M. Ross, *Introduction to Stochastic Dynamic Programming*. Academic Press, 1983.

[10] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 4–27, Jan. 2000.

[11] M. Feder and J. A. Catipovic, "Algorithms for joint channel estimation and data recovery-application to equalization in underwater communications," *IEEE J. Ocean. Eng.*, vol. 16, no. 1, pp. 42–55, Jan. 1991.

[12] W. Qian and J. A. Ritcey, "Spatial diversity equalization applied to underwater communications," *IEEE J. Ocean. Eng.*, vol. 19, no. 2, pp. 227–241, Apr. 1994.

[13] M. Stojanovic, "Recent advances in high-speed underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 21, no. 2, pp. 125–136, Apr. 1996.

[14] —, "Retrofocusing techniques for high rate acoustic communications," *Journal of the Acoustical Society of America*, vol. 117, no. 3, pp. 1173–1185, Mar. 2005.

[15] I. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Elsevier's Ad Hoc Networks*, vol. 3, no. 3, 2005.

[16] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 72–83, Jan. 2000.

[17] P. Casari, S. Marella, and M. Zorzi, "A comparison of multiple access techniques in clustered underwater acoustic networks," in *Proc. IEEE/OES OCEANS*, Aberdeen, Scotland, UK, Jun. 2007.

[18] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC Protocol for underwater acoustic networks," in *Proc. IEEE Oceans*, Singapore, Sep. 2006.

[19] X. Guo, M. Frater, and M. Ryan, "A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks," in *Proc. IEEE Oceans*, Singapore, Sep. 2006.

[20] A. F. Harris III, M. Stojanovic and M. Zorzi, "When underwater acoustic nodes should sleep with one eye open: idle-time power management in underwater sensor networks," in *Proc. ACM WUWNet*, Los Angeles, CA, USA, Sep. 2006, pp. 105–108.

[21] M. Stojanovic, "Optimization of a data link protocol for an underwater acoustic channel," in *IEEE Oceans*, June 2005.

[22] A. F. Harris III and M. Zorzi, "Energy-efficient routing protocol design considerations for underwater networks," in *Proc. IEEE SECON*, Jun. 2007.

[23] R. Urick, *Principles of Underwater Sound*. McGraw-Hill, 1983.

[24] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," in *Proc. ACM WUWNet*, Los Angeles, CA, USA, Sep. 2006, pp. 41–47.

[25] L. Berkhovskikh and Y. Lysanov, *Fundamentals of Ocean Acoustics*. Springer, 1982.

[26] M. Luby, "LT Codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, B.C., Canada, Nov. 2002.

[27] E. Hytiä, T. Tirronen and J. Virtamo, "Optimizing the Degree Distribution of LT Codes with an Importance Sampling Approach," in *RESIM 2006, 6-th International Workshop on Rare Event Simulation*, Bamberg, Germany, Oct. 2006.