Ad Hoc Networks 7 (2009) 791–802

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Efficient error recovery with network coding in underwater sensor networks ☆

Zheng Guo *, Bing Wang, Peng Xie, Wei Zeng, Jun-Hong Cui

Computer Science and Engineering Department, University of Connecticut, 371 Fairfield Way, Storrs, CT 06269, United States

ARTICLE INFO

Article history:
Received 8 October 2007
Received in revised form 22 July 2008
Accepted 28 July 2008
Available online 12 August 2008

Keywords:
Underwater sensor networks
Error recovery
Network coding
Multi-path routing

ABSTRACT

Before the wide deployment of underwater sensor networks becomes a reality, one challenge to be met is efficient error recovery in the presence of high error probability, long propagation delays and low acoustic bandwidth. We believe that network coding is a promising technique for this purpose due to Eq. (1) the computational capability of underwater sensor nodes, and Eq. (2) the broadcast nature of acoustic channels. In this paper, we propose an efficient error-recovery scheme that carefully couples network coding and multiple paths. Through an analytical study, we provide guidance on how to choose parameters in our scheme and demonstrate that the scheme is efficient in both error recovery and energy consumption. We evaluate the performance of this proposed scheme through simulation, and the simulation confirms the results from the analytical study.

1. Introduction

Over 70% of the surface of the earth is covered by water. Despite years of research, many critical underwater applications, such as oceanographic data collection, pollution monitoring, tactical surveillance applications, remain quite limited. The studies of [2–6] survey fundamental constraints, potential applications, challenges and future research directions in underwater environments. They point out that one ideal vehicle for these aquatic applications is underwater sensor network (UWSN) [4]. However, the characteristics of UWSNs, such as low bandwidth, long propagation delays and high error probability, are significantly different from those in terrestrial sensor networks. These unique characteristics pose a range of challenges [2–6]. One such challenge is efficient error recovery when using underwater acoustic channels. Under such severe network conditions, commonly used error-recovery techniques such as automatic repeat reQuest (ARQ) and forward error correction (FEC) become unsuitable (detailed in Section 2).

In a prior study, we demonstrate that network coding is a promising technique for error recovery in UWSNs [7]. The main idea of network coding [8,9] is that, instead of simply forwarding a packet, a node may encode several incoming packets into one or multiple outgoing packets. Network coding is suitable for UWSNs because Eq. (1) underwater sensor nodes are usually larger than land-based sensors and possess more computational capabilities [10]; and Eq. (2) the broadcast property of acoustic channels naturally renders multiple highly interleaved paths from a source to a sink. The computational power at the sensor nodes coupled with the multiple paths provides ample opportunity to apply network coding.

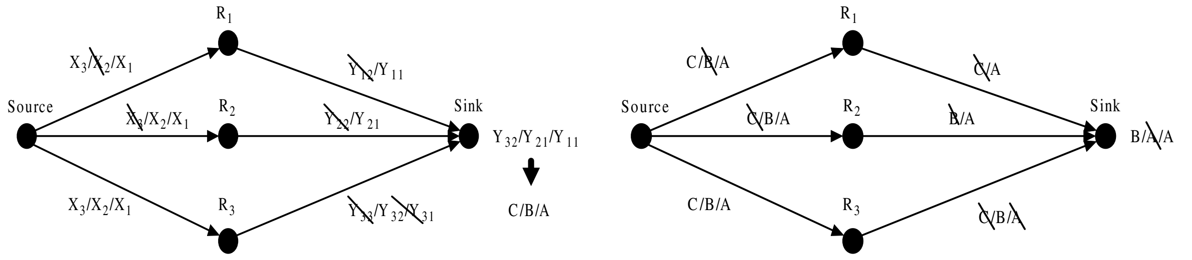
We now illustrate the benefits of network coding using a simple example in Fig. 1. Fig. 1a illustrates the result when using network coding. A source generates packets A, B and C, encodes these packets into X_1, X_2 and X_3, and then sends them to a sink.[1] These packets will reach relays

☆ A preliminary version of this paper [1] appeared in IFIP Networking 2007.

* Corresponding author. Tel.: +1 860 486 3665.
E-mail addresses: guozheng@engr.uconn.edu (Z. Guo), bing@engr.uconn.edu (B. Wang), xp@engr.uconn.edu (P. Xie), wei.zeng@engr.uconn.edu (W. Zeng), jcui@engr.uconn.edu (J.-H. Cui).

[1] A simple form of coding is that the encoded packets are copies of the original packets, e.g., $X_1 = A$, $X_2 = B$ and $X_3 = C$.

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.07.011



(a) Using network coding.

(b) Not using network coding.

Fig. 1. An example illustrating the benefits of using network coding in underwater sensor networks. A packet crossed out means that the packet is lost.

R_1, R_2 and R_3 simultaneously because of the broadcast property of the acoustic channel. Relay R_1 receives packets X_1 and X_3 successfully and encodes them into packets Y_{11} and Y_{12}. Similarly, relay R_2 encodes its incoming packets into packets Y_{21}, Y_{22}, and relay R_3 encodes its incoming packets into Y_{31}, Y_{32}, Y_{33}. The relays then forward the encoded packets to the sink. The sink receives three encoded packets Y_{11}, Y_{21}, and Y_{32}. When using a proper network coding scheme (e.g., random linear coding [11]), the sink can recover the three original packets with high probability. Fig. 1b illustrates the result when the relays simply forward the incoming packets without using network coding and discard duplicated packets. In this case, the sink only receives two distinct original packets.

In this paper, extending our preliminary work [7], we provide an in-depth study on using network coding in UWSNs. Our main contributions are as follows: Eq. (1) we propose an error-recovery scheme using network coding, and analytically study the performance of our scheme along with several other error-recovery schemes. Our analysis provides guidance on how to choose parameters in the proposed scheme and demonstrates that, of all schemes, our scheme is the most efficient in error recovery and energy consumption, and Eq. (2) we evaluate the performance of our scheme through simulation, and the simulation confirms the results from the analytical study.

The rest of the paper is organized as follows. We first discuss related work in Section 2. We then present the problem description and the propose an error-recovery scheme based on network coding in Sections 3 and 4 respectively. Section 5 analytically studies the performance of our scheme along with several other schemes. We next describe our evaluation methodology and evaluate the schemes through simulation in Section 6. Finally, Section 7 concludes the paper and presents future work.

2. Related work

Automatic repeat reQuest (ARQ) [12] and forward error correction (FEC) [13,14] are two conventional methods for error recovery. They, however, both have severe drawbacks when applied to UWSNs. ARQ-based schemes require the receiver to detect lost packets and then request the sender to retransmit packets. This may lead to a long delay before a packet is delivered successfully due to the slow propagation through acoustic channels. FEC-based schemes can be classified as end-to-end FEC and hop-by-hop FEC (e.g., [14]). These schemes proactively add redundant packets, so that the receiver (for hop-by-hop case) or the sink (for the end-to-end case) may successfully recover original packets, and hence eliminate the need for packet retransmissions. But the amount of redundancy needs to be sufficient to recover errors while conserving the limited battery power of sensor nodes. Determining the right amount of redundancy is, however, a challenging task due to the difficulty to obtain accurate error-rate estimates [4].

Due to the drawbacks of ARQ-based and FEC-based schemes, researchers have proposed other schemes to improve the robustness of sensor networks [10,15,16]. One scheme is multi-path forwarding [10,15], which uses redundant packets through multiple paths to improve packet delivery ratio. However, as we have seen in Fig. 1, multi-path forwarding alone is not sufficient because duplicated packets, which will be discarded directly, consume energy but do not provide any innovative information. Another scheme [16] uses multiple virtual sinks to provide error resilience: a source forwards packets to multiple virtual sinks using acoustic communication, then the virtual sinks forward the packets to the final destination using high-bandwidth wireless radio communication. This scheme requires a specialized delivery infrastructure. Our scheme applies to the single sink architecture and uses a collaborative coding scheme to fully utilize the scare resources in UWSNs.

Network coding is first proposed by Ahlswede et al. [8]. They use network coding to achieve the broadcast capacity of a multicast tree, which cannot be achieved by simply copying and forwarding packets. Afterwards, Li et al. prove that linear network coding is sufficient for the encoding functions [17]. Koetter et al. show how to find the coefficients of the linear coding and decoding functions [18]. Fragouli et al. present an instant primer on network coding in [9].

We apply network coding to UWSNs for reliable data transfer [1,7], and demonstrate that coupling network coding and multiple paths improves data delivery ratio and provides high energy efficiency. Lucani et al. propose a network-coding based method that relies on the implicit acknowledgement of previously transmitted packets to improve power consumption performance in UWSNs [19]. They consider a concatenated relay network and focus on the time to complete the transmissions of a given number of packets. In this paper, we focus on efficient error recovery using network coding in UWSNs with respect to the packet delivery ratio and energy consumption.

3. Problem description

We consider a source-sink pair in an underwater sensor network. The path (or multi-path) from the source to the sink is determined by a single-path (or multi-path) routing algorithm. We refer to the intermediate nodes on the path(s) as *relays*. We consider the basic single-path forwarding and several error-recovery schemes including end-to-end FEC, hop-by-hop FEC, multi-path forwarding and network coding. In single-path and multi-path forwarding, received packets are simply forwarded, without any coding. Single-path forwarding is a baseline scheme since it does not exploit any extra mechanism for error recovery. Multi-path forwarding recovers errors through redundant packets over multiple paths. FEC-based schemes use a single path from the source to the sink: end-to-end FEC encodes packets at the source and decodes them at the the sink; in hop-by-hop FEC, different from end-to-end FEC, each relay on the path decodes incoming packets, encodes the recovered packets, and then forwards them to the next hop. In network coding, a node encodes incoming packets into one or multiple outgoing packets (see Section 4).

A packet successfully received (under single or multi-path forwarding) or recovered (under FEC or network coding) is referred to as a *successfully delivered packet*. Since efficient error-recovery schemes for UWSNs must achieve high error-recovery rate and conserve sensor node energy simultaneously, we consider the following two metrics. The first metric is the fraction of the number of successfully delivered packets over the total number of packets from the source, referred to as *successful delivery ratio*, denoted as R. The second metric is the average energy consumed per packet from the source to the sink (including transmissions from the source and relays) normalized by the successful delivery ratio. We refer to this metric as *normalized energy consumption*, denoted as T. It represents the average energy required for a *successfully* delivered packet.

4. Using network coding in underwater sensor networks

We develop a scheme of applying network coding to UWSNs. To achieve a good balance between error recovery and energy consumption at the sensor nodes, the scheme carefully couples network coding and multiple paths (we refer this scheme as *network coding* in the rest of the paper). The reasons why we build our scheme on top of multiple paths instead of single-path are: Eq. (1) the broadcast property of the underwater acoustic channel naturally provides multiple paths, and Eq. (2) interleaved paths increase the opportunities of collaboration among relays and provide more innovative redundancy.

In the following, we first describe how to apply network coding (we use random linear coding [11] due to its simplicity) given a set of paths from a source to a sink. We then describe how to adjust the multiple paths or the amount of redundancy to improve the efficiency.

4.1. Network coding scheme

Packets from the source are divided into generations, with each generation containing K packets. The source linearly combines K packets in a generation using randomly generated coefficients. More specifically, let $X_1, \ldots, X_K$ denote the K packets in a generation. The source linearly combines these K packets to compute K' outgoing packets, denoted as $Y_1, Y_2, \ldots, Y_{K'}$ where $Y_i = \sum_{j=1}^{K} g_{ij} X_j$, $i = 1, 2, \ldots, K'$. The coefficient g_{ij} is picked randomly from a finite field F_{2^q}. The set of coefficients $(g_{i1}, \ldots, g_{iK})$ is referred to as the *encoding vector* for encoded packet Y_i [9] and is carried in the packet as overhead. We choose $K' \geqslant K$ since adding a small amount of redundancy at the source (e.g., $K' = K + 2$) reduces the impact of lost packets at the first hop (which cannot be recovered at later hops)[2] and improves error recovery at the sink [7].

A relay in forwarding paths stores incoming packets from different paths in a local buffer for a certain period of time, then linearly combines the buffered packets belonging to the same generation. Suppose a relay, r, receives M incoming packets, $X_1^r, \ldots, X_M^r$. Let $(f_{i1}, \ldots, f_{iK})$ denote the encoding vector carried by X_i^r, $i = 1, \ldots, M$. Since transmitting dependent packets is not useful for decoding at the sink, relay r computes M' outgoing packets, where M' is the rank of the coefficient matrix (f_{ij}), $i = 1, \ldots, M$, $j = 1, \ldots, K$. Therefore, $M' \leqslant \min(M, K)$. Let $Y_1^r, \ldots, Y_{M'}^r$ denote the outgoing packets, $Y_i^r = \sum_{j=1}^{M} h_{ij}^r X_j^r, i = 1, 2, \ldots, M'$, where h_{ij} is picked randomly from the finite field F_{2^q}. Let $(g_{i1}^r, \ldots, g_{iK}^r)$ denote the encoding vector of Y_i^r, $i = 1, \ldots, M'$. Then $g_{ij}^r = \sum_{l=1}^{M} h_{il}^r f_{lj}$.

When the sink receives K packets with linearly independent encoding vectors, it recovers the original packets by matrix inversion [9]. The complexity is $O(K^3)$.

4.2. Path or redundancy adaptation for network coding

The efficiency of network coding relies on the quality of the underlying paths determined by a multi-path routing algorithm. We next describe a property of a multi-path under which network coding is efficient (in both error recovery and energy consumption). Fig. 2 illustrates the process of transmitting a packet along a multi-path. The source broadcasts the packet to its downstream neighbors (nodes within its transmission range and in the forwarding paths), referred to as a *relay set*. Nodes in the relay set further forward the packet to their neighbors, forming another relay set. Intuitively, a multi-path suitable for network coding should contain a similar number of nodes in each relay set. This is because, a relay set with too few nodes may not provide sufficient redundancy; while a relay set with too many nodes wastes energy to provide more redundancy than what is necessary for error recovery.

[2] Note that there is only one source in our network setting. This means the first hop nodes only receive packets from the source, while other relay nodes may receive packets from multiple nodes in the previous hop. Thus, we intentionally introduce some redundancy at the source.

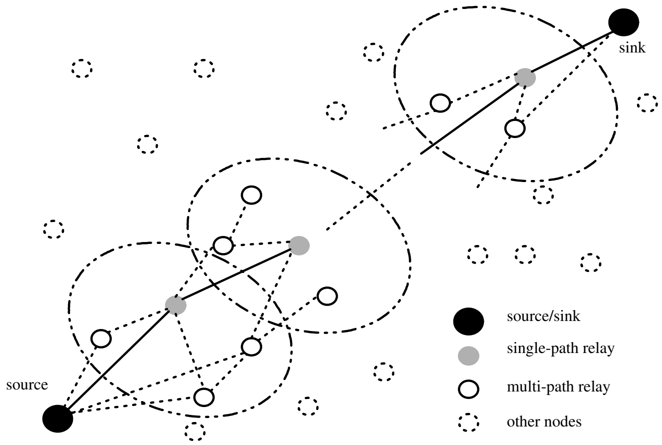


Fig. 2. Illustration of routing in a UWSN (nodes in a dashed circle form a relay set).

We develop two schemes to improve the efficiency of network coding. One scheme adjusts the routing paths, and the other one adjusts the amount of redundancy on each node. In both schemes, a node uses the number of its downstream neighbors to approximate the size of the downstream relay set. This is because the number of downstream neighbors can be easily estimated through localization services (e.g., [20]) and localized communication between a node and its neighbors, while the accurate size of the downstream relay set is difficult to estimate.

The first scheme requires sensor nodes to be equipped with transmitters which support multiple levels of transmission power [21]. A node selects a transmission power so that the estimated number of downstream neighbors is between N_l and N_u, where N_l and N_u are lower and upper thresholds respectively. We refer to this scheme as *transmission-range adaptation* and provide guidance on how to select N_l and N_u using our analytical results in Section 5.

The second scheme does not require nodes to have multiple levels of transmission power (i.e., each node has a fixed transmission range). Instead, in this scheme, a node adjusts the amount of redundancy that it injects into the network. More specifically, a node with fewer than N'_l downstream neighbors encodes more outgoing packets to increase the amount of redundancy. Similarly, a node with more than N'_u downstream neighbors encodes less outgoing packets to reduce the amount of redundancy (we only reduce the number of outgoing packets when the coefficient matrix has full rank of K to mitigate the risk of permanent information loss). We refer to this scheme as *redundancy adaptation* and investigate how to choose N'_l and N'_u using simulation in Section 6.

5. Analytical study

We now analytically study the performance of the various error-recovery schemes. Our goal is two-fold: Eq. (1) analytically compare the efficiency of the various schemes; and Eq. (2) provide guidance on how to choose parameters in network coding.

The setting of the analysis is illustrated in Fig. 2. For single-path based schemes, we assume a single path (marked by the solid line) with H hops, indexed from 1 to H. For multi-path based schemes, we assume H relay sets from the source to the sink, indexed from 1 to H. The sink is in the Hth relay set. Let N_i be the number of relays in the ith relay set. For simplicity, we assume that the relay sets do not intersect. Furthermore, a node in a relay set receives packets from all nodes in the previous relay set. Last, a node only uses packets forwarded from its previous relay set (i.e., packets received from nodes in the same relay set are discarded).

For all the schemes, we derive the normalized energy consumption, T, from the successful delivery ratio, R, as follows. The energy consumption is represented by the number of transmissions, which is proportional to the actual energy consumption. Consider an arbitrary packet (regardless of being successfully delivered or not), let T_i denote the average number of times that it is transmitted from the nodes in the previous relay set (or the source) to those in the ith relay set. Then

$$T = \frac{\sum_{i=1}^{H} T_i}{R}. \tag{1}$$

For simplicity, we assume that the acoustic channels have no fading and all sensor nodes transmit using the same power and bit rate. Thus, all links suffer the same bit error probability p_b. Let p be the probability that a packet has errors. Then $p = 1 - (1 - p_b)^L$ for independent bit errors and a packet of L bits.

5.1. Analysis of single-path forwarding

We now analyze single-path forwarding. Considering an arbitrary packet. The probability that this packet is transmitted successfully for i hop(s), p_i, is

$$p_i = (1 - p)^i, \quad 1 \leqslant i \leqslant H. \tag{2}$$

The successful delivery ratio, R, equals to the probability that the packet is transmitted successfully for H hops. That is, $R = p_H = (1 - p)^H$.

In the ith hop, the expected number of times that a packet is transmitted from the node in the previous hop, T_i, is

$$T_i = \begin{cases} 1, & i = 1, \\ p_{i-1}, & 2 \leqslant i \leqslant H. \end{cases} \tag{3}$$

After obtaining R and T_i, the normalized energy consumption is obtained from (1).

5.2. Analysis of end-to-end FEC

In end-to-end FEC, we consider an encoded block of $K+S$ packets, where K packets form an original block and a block is encoded to include S additional packets. As long as the sink receives K packets, it recovers the original block successfully. In end-to-end FEC, each packet in the block is delivered using single-path forwarding. Let R' and T' respectively denote the successful delivery ratio and the average number of transmissions for a packet in an encoded block. From the analysis of single-path forwarding, we have

$$R' = (1-p)^H, \tag{4}$$

$$T' = 1 + \sum_{i=2}^{H} (1-p)^{i-1}. \tag{5}$$

Because any K packets are sufficient to recover the original block, the successful delivery ratio is

$$R = \sum_{i=K}^{K+S} \cdot \binom{K+S}{i} R'^i (1-R')^{K+S-i}. \tag{6}$$

The normalized energy consumption is

$$T = \frac{\frac{K+S}{K} T'}{R}. \tag{7}$$

5.3. Analysis of hop-by-hop FEC

In hop-by-hop FEC, a relay decodes the incoming packets in the same encoded block to recover the original packets. If the decoding is successful, then it encodes the recovered packets into an encoded block of $K+S$ packets again, and forwards it to its next hop neighbor. Otherwise, it discards the incomplete block. Hop-by-hop FEC is similar to single-path forwarding when regarding the entire block as a packet. So the probability to lose a block in one hop, p', is

$$p' = \sum_{i=0}^{K-1} \binom{K+S}{i} (1-p)^i p^{K+S-i} \tag{8}$$

and the number of transmissions in the ith hop is

$$T_i = \begin{cases} \frac{K+S}{K}, & i = 1, \\ \frac{K+S}{K}(1-p')^{i-1}, & 2 \leqslant i \leqslant H. \end{cases} \tag{9}$$

The successful delivery ratio and normalized energy consumption are

$$R = (1-p')^H, \quad T = \frac{\sum_{i=1}^{H} T_i}{R}. \tag{10}$$

5.4. Analysis of multi-path forwarding

Consider an arbitrary packet P. Let α_i be the probability that a node in the ith relay set receives packet P. Let $\alpha_{i,n}$ be the probability that n nodes in the ith relay set receive packet P, $n = 0, \ldots, N_i$. Assume that packet losses are independent. Then

$$\alpha_i = \begin{cases} 1-p & i = 1, \\ \sum_{n=0}^{N_{i-1}} \alpha_{i-1,n}(1-p^n), & 2 \leqslant i \leqslant H. \end{cases} \tag{11}$$

This is because, for a node in the first relay set, the probability that it receives packet P from the source is $1-p$; when $i \geqslant 2$, a node in the ith relay set receives packet P when it receives at least one copy of this packet from the $(i-1)$th relay set. Assume that packet transmissions to nodes in a relay set are independent. Then

$$\alpha_{i,n} = \binom{N_i}{n} \alpha_i^n (1-\alpha_i)^{N_i - n}, \quad n = 0, \ldots, N_i. \tag{12}$$

Since packet P is an arbitrary packet and the sink is in the Hth set, we have $R = \alpha_H$. The above results indicate that α_H can be obtained in the following manner. We first obtain $\alpha_{1,n}$ from α_1 (of value $1-p$), and then obtain α_2 using $\alpha_{1,n}$. This process continues until eventually α_H is obtained. Since a node forwards packet P at most once, we have

$$T_i = \begin{cases} 1, & i = 1, \\ \alpha_{i-1} N_{i-1}, & 2 \leqslant i \leqslant H. \end{cases} \tag{13}$$

After obtaining R and T_i, we calculate the normalized energy consumption T from (1).

5.5. Analysis of network coding

Consider an arbitrary generation of K packets. Under linear random coding, when a sink receives at least K packets in the generation, the probability that it recovers the K original packets is high for a sufficiently large finite field [11]. Therefore, for simplicity, we assume that the sink recovers the K original packets as long as it receives at least K packets in the generation. We do not differentiate between the nodes in the same relay set. Let $\beta_{i,k}$ be the probability that a node in the ith relay set receives k packets (when $0 \leqslant k < K$) or at least k packets (when $k = K$) from all nodes in the previous relay set, $1 \leqslant i \leqslant H$. Since the sink is in the Hth relay set and the generation is arbitrary, we have $R = \beta_{H,K}$.

We next derive $\beta_{i,k}$, $1 \leqslant i \leqslant H$, $0 \leqslant k \leqslant K$. The nodes in the first relay set receive packets from the source. Therefore

$$\beta_{1,k} = \begin{cases} (K'k)(1-p)^k p^{K'-k}, & 0 \leqslant k < K, \\ 1 - \sum_{j=0}^{K-1} \beta_{1,j} & k = K. \end{cases} \tag{14}$$

where $K' \geqslant K$ is the number of encoded packets from the source.

For $i \geqslant 1$, $0 \leqslant k < K$, we obtain $\beta_{i+1,k}$ as follows. We index the nodes in the ith relay from 1 to N_i. Let $\gamma_{i,j,k}$ denote the probability that a node in the ith relay set receives k packets from the jth node in the previous relay set,

$1 \leqslant i \leqslant H$, $1 \leqslant j \leqslant N_{i-1}$, $0 \leqslant k < K$. Since each relay transmits no more than K packets, we have

$$\gamma_{i,j,k} = \sum_{n=k}^{K} \beta_{i-1,k} \binom{n}{k} (1-p)^k p^{n-k}. \tag{15}$$

For a node in the $(i+1)$th set, let k_j be the number of packets that it receives from the jth node in the previous relay set. To obtain $\beta_{i+1,k}$, we need to consider all combinations of k_j's such that $\sum_{j=1}^{N_i} k_j = k$, $k_j = 0, \ldots, k$. That is

$$\beta_{i+1,k} = \sum_{k_j=0,\ldots,k \text{ s.t. } \sum_{j=1}^{N_i} k_j = k} \prod_{j=1}^{N_i} \gamma_{i+1,j,k_j}. \tag{16}$$

For a small generation size K, the above quantity is easy to compute. We use small K (e.g., $K = 3$) since our study [7] indicates that it is sufficient to achieve good performance using small K (also confirmed by simulation in the settings of Section 6). We obtain $\beta_{i+1,K}$ from $\beta_{i+1,k}$, $0 \leqslant k < K$ as

$$\beta_{i+1,K} = 1 - \sum_{k=0}^{K-1} \beta_{i+1,k}. \tag{17}$$

From the above, we calculate $R = \beta_{H,K}$ as follows. We first obtain $\beta_{1,k}$, which is used to compute $\gamma_{2,j,n}$ and $\beta_{2,k}$, $0 \leqslant k \leqslant K$. This process continues until eventually $\beta_{H,K}$ is obtained. Since a relay transmits no more than K packets, we have

$$T_i = \begin{cases} K'/K, & i = 1, \\ \frac{N_{i-1}}{K} \sum_{k=0}^{K} k\beta_{i-1,k}, & 2 \leqslant i \leqslant H. \end{cases} \tag{18}$$

After obtaining R and T_i, we calculate the normalized energy consumption T from (1).

5.6. Numerical results

We next compare the various schemes based on the analysis. The bit error rate is in the range of 10^{-4} to 1.5×10^{-3} to account for potentially high loss rate in underwater sensor networks. For network coding, a generation contains three packets (e.g., $K = 3$). The source transmits $K' = 5$ packets. For multi-path forwarding and network coding, we set the number of relay sets, H, to 7 or 9, and assume all relay sets contain the same number of nodes, i.e., $N_i = N$, $i = 1, \ldots, H$. Similarly, for single-path forwarding and FEC, we set the number of hops from the source to the sink to 7 or 9. For FEC, each block contains three packets (same as the generation size in network coding) and each block is encoded into $3N$ packets to keep the same amount of redundancy as that in multi-path based schemes, where each relay set contains N nodes.

Fig. 3a and b plots respectively the successful delivery ratio and normalized energy consumption for various schemes, where $H = 9$ and $N = 3$. We observe that network coding outperforms the other schemes: it achieves the highest successful delivery ratio for the range of bit error rate and the lowest normalized energy consumption for most of the bit error rates. Multi-path forwarding achieves a similar normalized energy consumption and a lower successful delivery ratio than network coding. Hop-by-hop FEC achieves much lower successful delivery ratio than network coding under high bit error rates. For single-path forwarding and end-to-end FEC, the successful delivery ratio decreases and the normalized energy consumption increases dramatically as the bit error rate increases, indicating that they are not suitable for high error-rate UWSNs.

In the above, we fixed $N = 3$ and $H = 9$. We next explore the impact of these two parameters. Fig. 4a demonstrates the impact of N (the number of nodes in a relay set) on the performance of multi-path forwarding and network coding. We observe that when N decreases from 3 to 2, the successful delivery ratios of both schemes drop sharply. This implies that a node should have three downstream neighbors for efficient error recovery in multi-path based schemes (under our assumptions, each node has N downstream neighbors). Fig. 4b demonstrates the impact of H (the number of hops on a path) on the performance of multi-path forwarding, network coding and hop-by-hop FEC. We observe that network coding and multi-path forwarding achieve similar performance for $H = 7$ and 9, indicating that they are insensitive to the length of the path. The performance of hop-by-hop FEC, however, degrades significantly when increasing the path length from 7 to 9.

In summary, the analytical results above indicate that multi-path based schemes are more suitable for UWSNs. Hop-by-hop FEC outperforms other single-path based schemes. However, its performance is still sensitive to both bit error rates and path length, and hence the amount of redundancy needs to be carefully selected according to these two parameters.

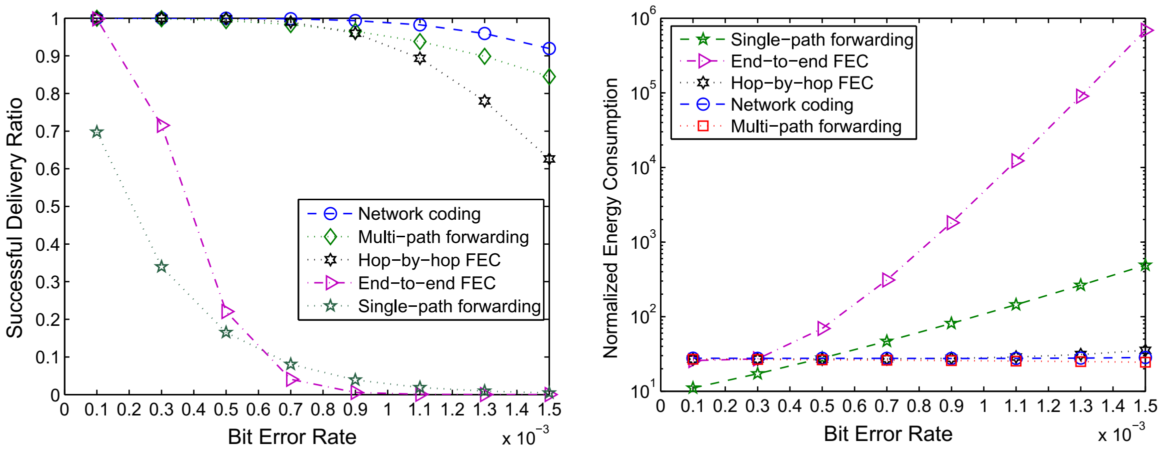
6. Performance evaluation

We evaluate the performance of the various error-recovery schemes using simulation in a wide range of settings. The simulation is through two simulators those are complementary to each other. We next describe these two simulators and the simulation setting. Afterwards, we detail the evaluation results. Our focus is on multi-path based schemes (i.e., multi-path forwarding and network coding) as our analysis has shown that single-path based schemes are not suitable for UWSNs.

6.1. Network simulators

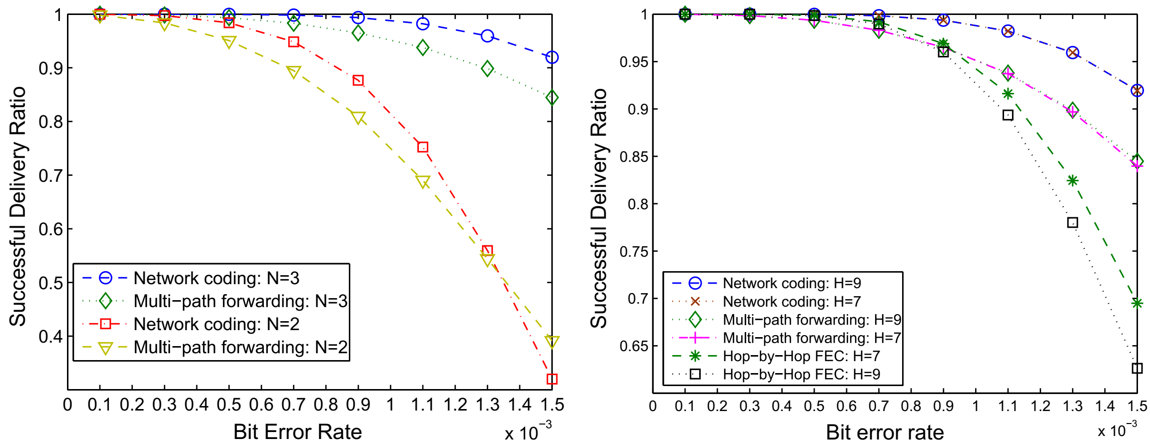
We have implemented two simulators, one built using Matlab and the other built on top of ns2. These two simulators are complementary to each other: the Matlab-based simulator is simpler and more flexible, while the ns2-based simulator simulates a more realistic underwater environment. We next describe these two simulators in more detail.

The Matlab-based simulator simulates a simplified environment: it uses centralized packet scheduling and forwarding to avoid packet collisions; it assumes the same amount of time to transmit a packet from a node to all its neighbors; and it only considers energy consumption during packet transmission and uses the number of transmissions to represent the amount of energy consumption.



(a) Successful delivery ratio. (b) Normalized energy consumption.

Fig. 3. Numerical results, $H = 9$, $N = 3$.



(a) $H = 9$. (b) $N = 3$.

Fig. 4. Numerical results: (a) impact of N (the number of nodes in a relay set) on the performance of multi-path based schemes and (b) impact of H (length of a path).

Although simplistic, this simulator has two important advantages. First, it makes similar assumptions as those in the analysis, and hence we use it to validate the analytical results. Secondly, it is more flexible. For instance, it can easily support transmission range adaptation, which is not supported by the current version of the ns2-based simulator.

The ns2-based simulator is an underwater network simulation package, called *Aqua-Sim*, developed by the UWSN Lab at the University of Connecticut [22]. It supports three dimensional infrastructure and uses erasure channel as the physical channel model. This simulator does not explicitly implement the physical channel fading and noise, while it only considers an erasure channel with a packet loss probability, which follows Bernoulli distribution for each packet on each link. However, this simulator simulates propagation delays (with acoustic wave speed), link bandwidth, energy consumption (in Joules), shared medium access, collisions and real-time packet scheduling process. The shared medium access protocol works as follows. When a node has packets to broadcast, it senses the channel first. If the channel is clear, it will broadcast these packets immediately. Otherwise, it will delay the broadcast for a random period chosen from the backoff interval. A node delays a broadcast at most 4 times before dropping the packets. The power consumption parameters follow a commercial product, UWM1000 modem from LinkQuest [23]. That is, the power consumptions in the transmission, receiving and sleeping modes are 2 W, 0.75 W and 0.008 W, respectively. The energy consumption is the sum of the energy consumed in all the three modes.

6.2. Simulation settings

In the simulation, underwater sensor nodes are distributed in a 3D cubic area of 1 km × 1 km × 1 km. We focus on static networks. The source and sink are deployed respectively at the bottom corner and surface corner, on the diagonal of the cube. Other sensor nodes are randomly deployed in the cube. Each node has a transmission range of 300 m and a data rate of 10 kbps.

The multiple paths from the source to the sink are determined by vector-based forwarding (VBF) with a 150 m routing pipe [10]. In VBF, a routing pipe is a pipe centered around the vector from the source to the sink. Nodes inside the routing pipe are responsible for routing packets from the source to the sink; nodes outside the routing pipe simply discard the received packets. We set the original data packet size to 50 bytes. For network coding, each generation contains $K = 3$ packets, the source outputs $K' = 5$ packets, and each relay outputs at most three packets for each encoded generation. The reason why we choose these parameters is that they achieve a good performance in both delivery ratio and normalized energy consumption (the impact of various parameters is detailed in Section 6.3.2). We choose a finite field of F_{2^8} [11]. Therefore, each packet is 53 bytes long (including a 3-byte encoding vector) under network coding.

The source generates data at a constant rate of 1 packet per second and each simulation runs for 350 s. In the network coding scheme, the source waits for K packets to form a generation, and then sends them out back to back. Each relay node delays a random time interval (between 3 and 4 s) to wait for the arrival of packets belonging to the same generation, then generates new independent encoded packets and forwards them to the next hop. Late packets of the generation will be simply discarded.

We investigate two types of network deployment: *grid random deployment* and *uniform random deployment*. In grid random deployment, the target area is divided into grids with a number of nodes randomly deployed in each grid. More specifically, the target area is divided into 125 grids, each of 200 m $\times$ 200 m $\times$ 200 m. Each grid contains two nodes that are randomly distributed in the grid. In uniform random deployment, all nodes are uniformly randomly deployed in the area. Clearly, grid random deployment covers the area more evenly than uniform random deployment, while uniform random deployment is easier during implementation.

6.3. Performance under grid random deployment

Under grid random deployment, each node covers a similar number of downstream neighbors when using the same transmission power. Therefore, there is no need for transmission power or redundancy adaptation. In the following, we first present the results on successful delivery ratio and normalized energy consumption, and then investigate the impact of several parameters.

6.3.1. Successful delivery ratio and normalized energy consumption

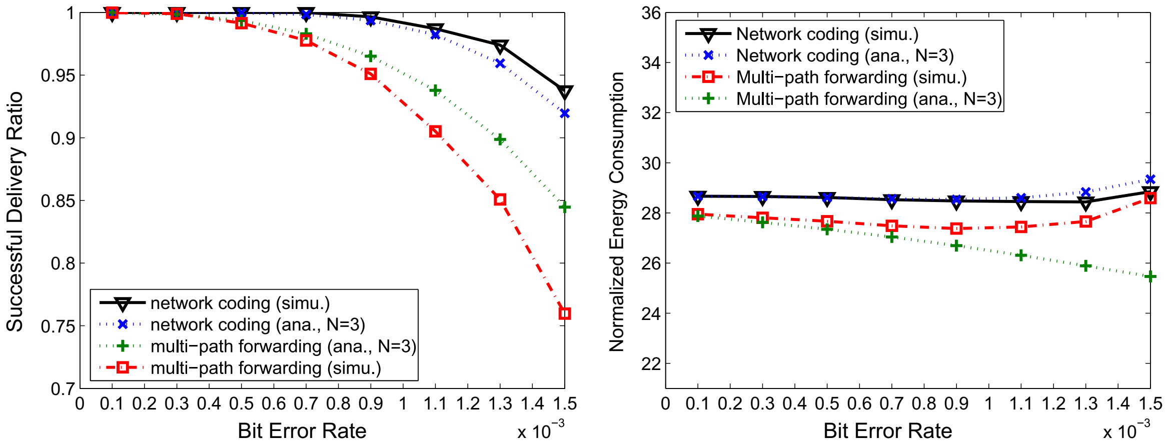
The analytical results in Section 5 indicate that each node should have three downstream neighbors for efficient error recovery in multi-path based schemes. We therefore set the transmission range to 300 m for each node and set the pipe radius to 150 m so that each node has 3–4 downstream neighbors. We first validate the analytical results using simulation results from the Matlab-based simulator, and then present the results from *Aqua-Sim*.

Fig. 5a and b plots the successful delivery ratio and normalized energy consumption for network coding and multi-path forwarding. The results from both the analysis and the Matlab-based simulator are plotted in the figures. For network coding, the simulation results are very close to those from the analysis, indicating that the analysis provides a good approximation. For multi-path forwarding, the successful delivery ratios from the analysis are slightly higher (no more than 8%) than those from the simulation. This might be because our analysis assumes that a node can hear from all the nodes in its previous relay set, which leads to an overestimate of the successful delivery ratio. This overestimation also contributes to slightly lower normalized energy consumption from the analysis than that from the simulation. In general, we observe that our analytical results match those from the Matlab-based simulator.

We now look at simulation results from *Aqua-Sim*, which simulates a more realistic underwater environment. Fig. 6a and b plots the successful delivery ratio and normalized energy consumption (in Joules) for network coding and multi-path forwarding. The 95% confidence intervals are from 25 runs. For low packet loss rates (below 0.1), multi-path forwarding achieves a slightly higher successful delivery ratio than network coding; for high loss rates, network coding achieves a significantly higher successful delivery ratio than multi-path forwarding. The reason why multi-path forwarding outperforms network coding under low loss rates is as follows. When the packet loss rate is low, multi-path forwarding provides sufficient amount of redundancy for error recovery, and hence achieves a high successful recovery ratio. In network coding, a node waits for a generation of packets and then sends them back to back. This leads to more bursty packet transmissions. Moreover, network coding recovers and sends more packets. These two reasons cause more collisions (confirmed by our simulation) and a lower successful delivery ratio. Under the same packet loss rate, the successful delivery ratios under both multi-path forwarding and network coding are lower than those from the analysis and Matlab-based simulator. This is not surprising since *Aqua-Sim* takes account of many constraints in the real environment, such as packet collisions, link bandwidth and long propagation delays. Fig. 6b plots the normalized energy consumption (In Joules) for network coding and multi-path forwarding. For low packet loss rates, multi-path forwarding outperforms network coding, while for high loss rates, network coding outperforms multi-path forwarding. This is consistent with the results on successful delivery ratio in Fig. 6a. In general, we observe that results from *Aqua-Sim* exhibit similar trends as those from the analysis, indicating that the analysis can provide valuable guidance on the choice of parameters as we shall see in Section 6.3.2.

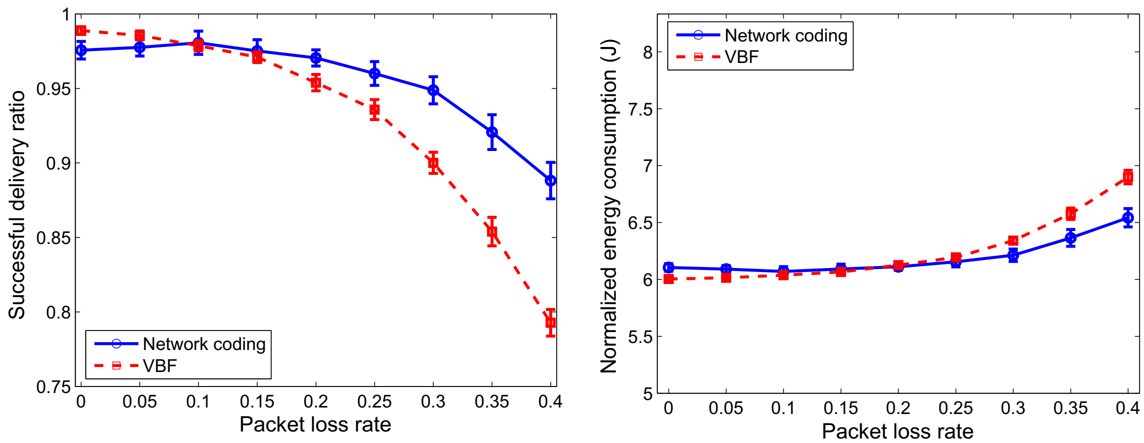
6.3.2. Impact of several parameters

We now explore the impact of several important parameters on the performance of the network-coding



(a) Successful delivery ratio. (b) Normalized energy consumption.

Fig. 5. Grid random deployment: analytical results and simulation results from the Matlab-based simulator: (a) successful delivery ratio and (b) normalized energy consumption.



(a) Successful delivery ratio. (b) Normalized energy consumption.

Fig. 6. Grid random deployment: simulation results from *Aqua-Sim*: (a) successful delivery ratio and (b) normalized energy consumption.

based scheme under the grid random deployment. All the results below are from *Aqua-Sim*. The packet loss rate is fixed to 0.35.

The number of downstream neighbors. We now demonstrate that it is indeed important for a node to have 3–4 downstream neighbors for efficient network coding, as indicated by the analytical results. For this purpose, we either fix the transmission range to 300 m and vary the pipe radius, or fix the pipe radius to 150 m and vary the transmission range. The results are plotted in Fig. 7a and b respectively. In both cases, we observe that a good balance between successful delivery ratio and normalized energy consumption is achieved when the transmission range is 300 m and the pipe radius is 150 m, i.e., when a node has 3–4 downstream neighbors. This demonstrates that our analysis, although carried out under simplifying assumptions, provide accurate guidance for the choice of parameters.

First-hop redundancy. We now explore the impact of first-hop redundancy. Fig. 8a plots the performance results when the source adds different amount of redundancy. When not adding any redundancy, the successful delivery ratio is very low because a packet lost in the first hop cannot be recovered later and cause an entire generation irrecoverable. When adding redundancy at the source, the successful delivery ratio increases significantly. This demonstrates the importance of first-hop redundancy. However, too much redundancy is not necessary because it will cause more packets forwarded along the path and lead to higher energy consumption.

Generation size. Generation size is an important parameter because it determines the amount of overhead in an encoded packet (i.e., the coding vector). Fig. 8b plots the results when varying the number of packets in a generation from 1 to 5. When a generation only contains a single packet, network coding reduces to multi-path forwarding.

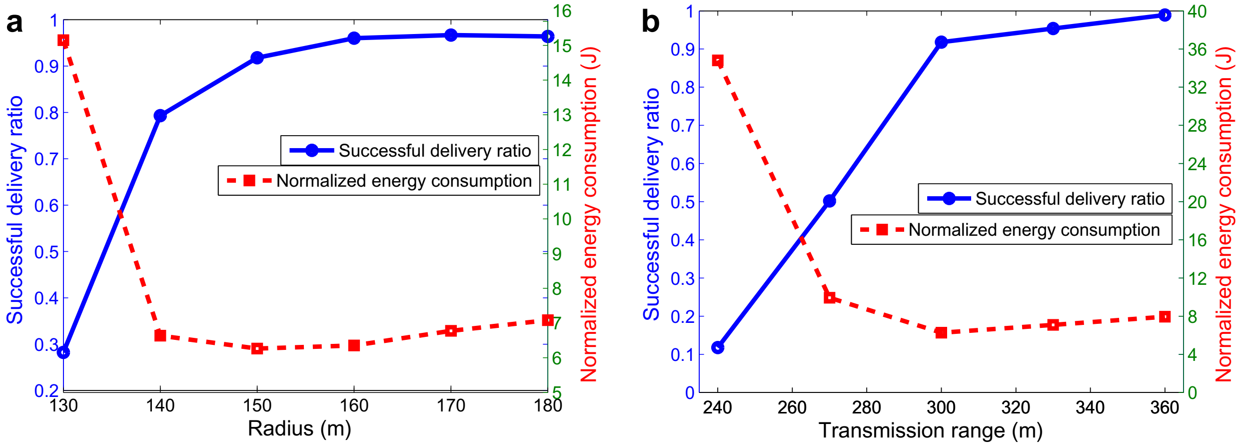
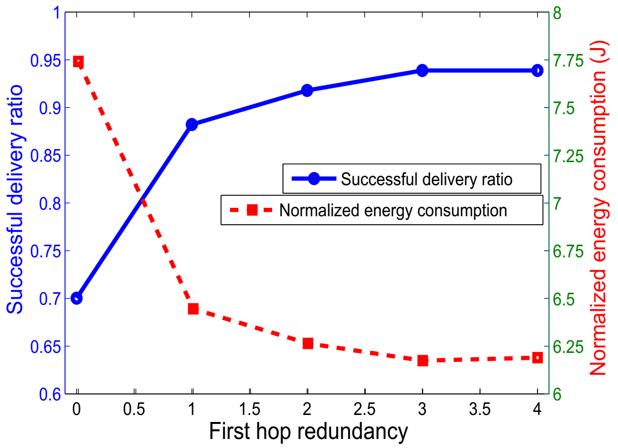
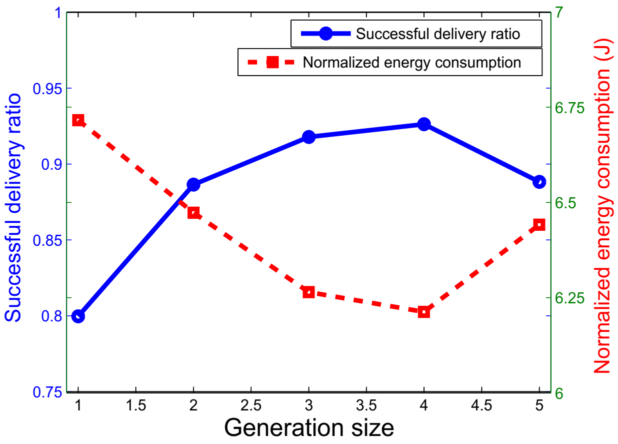


Fig. 7. Grid random deployment: impact of the number of downstream neighbors on the performance of the network-coding based scheme: (a) vary pipe radius, transmission range is 300 m and (b) vary transmission range, pipe radius is 150 m.



(a) Impact of first-hop redundancy.



(b) Impact of generation size.

Fig. 8. Grid random deployment: impact of first-hop redundancy and generation size on the performance of the network-coding based scheme.

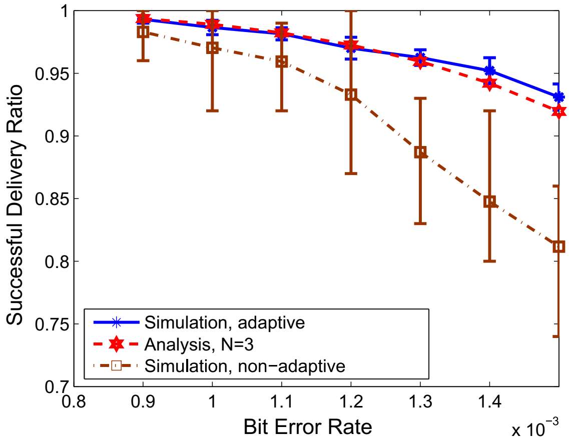
When increasing the generation size, the successful delivery ratio increases and the normalized energy consumption decreases. However, a very large generation is not desirable since the sink may not be able to receive sufficient number of packets to recover a generation.

6.4. Performance under uniform random deployment

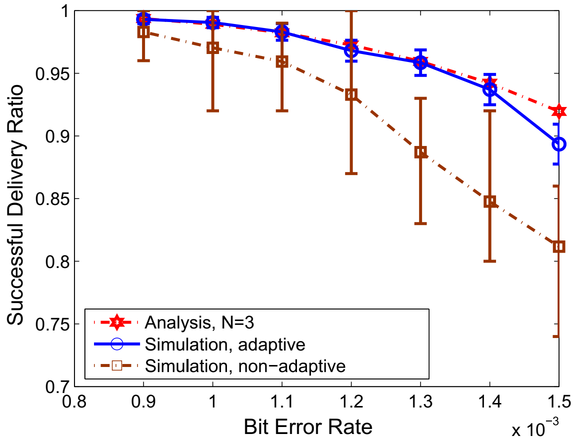
Under uniform random deployment, we find that using the same transmission range for all the nodes cannot ensure 3–4 downstream neighbors for each node. We therefore allow a node to adjust its transmission range or the amount of redundancy that it injects into the network based on the network topology. In the following, we only present the results from the Matlab-based simulator (the current version of *Aqua-Sim* does not support transmission range adaptation). Our focus below is on the benefits from transmission-range adaptation and redundancy adaptation.

We first present the results for transmission-range adaptation. We set the pipe radius to 150 m. Each node adapts its transmission range to have 3–4 downstream neighbors (the transmission range of each node varies from 100 to 400 m). Fig. 9a plots the successful delivery ratio using network coding. The confidence intervals are obtained from 20 simulation runs. We observe that transmission-range adaption achieves a similar successful delivery ratio as the analysis when $N = 3$. This indicates that transmission-range adaption is effective for error recovery. For comparison, we also plot the successful delivery ratio when all nodes uses a transmission range of 300 m, which is significantly lower than that applying transmission-range adaptation. Furthermore, the normalized energy consumption with transmission-range adaptation is lower than that when all nodes use the same transmission range (not plotted).

We next present the results when all nodes use the same transmission range of 300 m but adjust the amount of redundancy according to the number of its downstream neighbors: a node adds one more outgoing packet when it has fewer than three downstream neighbors and removes



(a) Transmission-range adaption.



(b) Redundancy adaption.

Fig. 9. Uniform random deployment: the benefits of transmission-range adaptation and redundancy adaptation in network coding.

an outgoing packet when it has more than six downstream neighbors. From Fig. 9b, we observe that this adaption achieves a similar successful delivery ratio as the analysis when $N = 3$ with only slightly higher normalized energy consumption (not plotted). The above results demonstrate that adjusting redundancy is also helpful for efficient error recovery using network coding.

7. Conclusions and future work

In this paper, we propose an efficient error recovery scheme that carefully couples network coding and multiple paths in UWSNs. Then we analytically study the performance of our scheme along with several other error recovery schemes. The analysis provides guidance on how to choose parameters in the proposed scheme and demonstrates that our scheme is the most efficient among multiple schemes. Last, we evaluate the performance of various schemes through simulation. The simulation results confirm the analytical study that our scheme is efficient in both error recovery and normalized energy consumption in UWSNs.

As future work, we are pursuing in the following directions: Eq. (1) using network coding in multicast applications in UWSNs, e.g., command distribution or software update from one source to all other nodes, which require elegant coordination among the multiple receivers as well as the relay nodes, Eq. (2) applying network coding for information collection in UWSNs, in which one important issue is generation management to avoid large overhead of coding vectors, and Eq. (3) treating the underwater sensor network as a distributed database and employing network coding for efficient distributed storage and retrievals.

References

[1] Z. Guo, B. Wang, J.-H. Cui, Efficient error recovery using network coding in underwater sensor networks, in: Proceedings of IFIP Networking'07, Atlanta, Georgia, USA, May 2007, pp. 227–238.
[2] E.M. Sozer, M. Stojanovic, J.G. Proakis, Underwater acoustic networks, IEEE Journal of Oceanic Engineering 25 (2000) 72–83.
[3] I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges, Ad Hoc Networks Journal, (Elsevier) 3 (2005) 257–279.
[4] J.-H. Cui, J. Kong, M. Gerla, S. Zhou, Challenges: building scalable mobile underwater wireless sensor networks for aquatic applications, IEEE Network, Special Issue on Wireless Sensor Networking 20 (2006) 12–18.
[5] J. Heidemann, W. Ye, J. Wills, A. Syed, Y. Li, Research challenges and applications for underwater sensor networking, in: Proceedings of the IEEE Wireless Communications and Networking Conference, Las Vegas, Nevada, USA, April 2006, pp. 228–235.
[6] J. Partan, J. Kurose, B.N. Levine, A survey of practical issues in underwater networks, in: Proceedings of ACM WUWNet'06, Los Angeles, CA, USA, September 2006.
[7] Z. Guo, P. Xie, J.-H. Cui, B. Wang, On applying network coding to underwater sensor networks, in: Proceedings of ACM WUWNet'06, Los Angeles, CA, USA, September 2006.
[8] R. Ahlswede, N. Cai, S.R. Li, R.W. Yeung, Network information flow, IEEE Transactions on Information Theory 46 (2000).
[9] C. Fragouli, J.-Y.L. Boudec, J. Widmer, Network coding: an instant primer, ACM SIGCOMM Computer Communication Review (2006).
[10] P. Xie, J.-H. Cui, L. Lao, VBF: vector-based forwarding protocol for underwater sensor networks, in: Proceedings of IFIP Networking'06, Coimbra, Portugal, May 2006, pp. 228–235.
[11] T. Ho, R. Koetter, M. Medard, D.R. Karger, M. Effros, The benefits of coding over routing in a randomized setting, in: Proceedings of IEEE International Symposium on Information Theory (ISIT), 2003.
[12] L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, third ed., May 2003.
[13] G.C. Clark Jr., J.B. Cain, Error-Correction Coding for Digital Communications, Plenum Press, New York, 1981.
[14] P. Xie, J.-H. Cui, An FEC-based reliable data transport protocol for underwater sensor networks, in: Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN'07), Honolulu, Hawaii, USA, August 2007.
[15] N. Nicolaou, A. See, P. Xie, J.-H. Cui, D. Maggiorini, Improving the robustness of location-based routing for underwater sensor networks, in: Proceedings of IEEE OCEANS'07, Aberdeen, Scotland, June 2007.
[16] W.K. Seah, H. Tan, Multipath virtual sink architecture for underwater sensor networks, in: Proceedings of MTS/IEEE OCEANS2006 Asia Pacific Conference, Singapore, May 2006.
[17] S.-Y.R. Li, R.W. Yeung, N. Cai, Linear network coding, IEEE Transactions on Information Theory 49 (2003) 371–381.
[18] R. Koetter, M. Medard, An algebraic approach to network coding, IEEE/ACM Transactions on Networking 11 (2003) 782–795.
[19] D. Lucani, M. Medard, M. Stojanovic, Network coding schemes for underwater networks: the benefits of implicit acknowledgement, in: Proceedings of ACM WUWNet'07, Montreal, Quebec, Canada, September 2007.
[20] V. Chandrasekhar, W.K. Seah, Y.S. Choo, H.V. Ee, Localization in underwater sensor networks – survey and challenges, in:

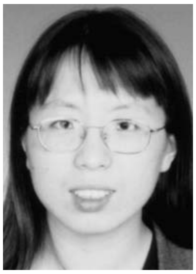
Proceedings of ACM WUWNet'06, Los Angeles, California, USA, September 2006.
[21] J. Wills, W. Ye, J. Heidemann, Low-power acoustic modem for dense underwater sensor networks, in: Proceedings of ACM WUWNet'06, Los Angeles, CA, September 2006.
[22] Aqua-Sim: An NS2 based Underwater Sensor Network Simulator, Underwater Sensor Network (UWSN) Lab, University of Connecticut, URL: http://uwsn.engr.uconn.edu/aquasim.tar.gz.
[23] LinkQuest Inc., Underwater Acoustic Modem Models (http://www.link-quest.com/html/uwm_hr.pdf).

Zheng Guo received his bachelor degree in Electronic Engineering from University of Science and Technology of China (USTC) in 2005. Currently he is pursuing a Ph.D. degree the Computer Science and Engineering Department at the University of Connecticut. His research interests are network coding, channel coding, routing and delay/disruption tolerant network in the area of underwater sensor network (UWSN) under the instructions from Prof. Bing Wang and Prof. Jun-Hong Cui.

Bing Wang received her B.S. degree in Computer Science from Nanjing University of Science and Technology, China in 1994, and M.S. degree in Computer Engineering from Institute of Computing Technology, Chinese Academy of Sciences in 1997. She then received M.S. degrees in Computer Science and Applied Mathematics, and a Ph.D. in Computer Science from the University of Massachusetts, Amherst in 2000, 2004, and 2005 respectively. Afterwards, she joined the Computer Science and Engineering Department at the University of Connecticut as an assistant professor. Her research interests are in Computer Networks, Multimedia, and Distributed Systems. More specifically, she is interested in topics on Internet technologies and applications, wireless and sensor networks, overlay networks, content distribution, network management and measurement, network modeling and performance evaluation. She is a member of ACM, ACM SIGCOMM, IEEE, IEEE Computer Society, and IEEE Communications Society.

Peng Xie received his B.E. and M.S. degrees in Computer Engineering from Harbin Institute of Technology (HIT), China, in 1990 and 1995 respectively, and his Ph.D. degree in the Computer Science and Engineering Department from the University of Connecticut in 2008, majoring in computer networks. He is currently a research scientist at Intelligent Automation Inc. His expertise includes research and development on network protocols, network and system security, network management, and distributed systems.

Wei Zeng received the bachelor and master degrees in Computer Science and Engineering from South China University of Technology, Guangzhou, China, in 2000 and 2003. Currently she is a Ph.D. student in the Computer Science and Engineering Department at the University of Connecticut, working with Professor Bing Wang. She is doing researches about network diagnosis, network measurement and network management for the Internet and sensor networks.



Jun-Hong Cui received her B.S. degree in Computer Science from Jilin University, China in 1995, her M.S. degree in Computer Engineering from Chinese Academy of Sciences in 1998, and her Ph.D. degree in Computer Science from UCLA in 2003. Currently, she is on the faculty of the Computer Science and Engineering Department at University of Connecticut. Her research interests cover the design, modeling, and performance evaluation of networks and distributed systems. Recently, her research mainly focuses on exploiting the spatial properties in the modeling of network topology, network mobility, and group membership, scalable and efficient communication support in overlay and peer-to-peer networks, algorithm and protocol design in underwater sensor networks. She is actively involved in the community as an organizer, a TPC member, and a reviewer for many conferences and journals. She is a guest editor for ACM MCCR (Mobile Computing and Communications Review) and Elsevier Ad Hoc Networks. She co-founded the first ACM International Workshop on UnderWater Networks (WUWNet'06), and she is now serving as the WUWNet steering committee chair. She is a member of ACM, ACM SIGCOMM, ACM SIGMOBILE, IEEE, IEEE Computer Society, and IEEE Communications Society. Her email address is jcui@cse.uconn.edu.