Network coding to combat packet loss in underwater networks

Mandar Chitre
Department of Electrical & Computer Engineering
& ARL, Tropical Marine Science Institute
National University of Singapore
mandar@arl.nus.edu.sg

Wee-Seng Soh
Department of Electrical & Computer Engineering
National University of Singapore
weeseng@nus.edu.sg

Abstract

Channel variability and a high level of ambient noise lead to significant probability of packet loss in many underwater networks. Techniques based on acknowledgements and re-transmissions (such as ARQ) can be used to build robust networks over the unreliable links between underwater nodes. An alternative solution based on erasure codes can also be used to combat the packet loss. However, both solutions rely on a node re-transmitting information originating at that node. We propose an alternative solution based on network coding, where nodes transmit packets which are composed partially from information originating at that node, and partially from information received by that node from other nodes. The intuition behind this solution is to effectively route the information over good paths in the network rather than to simply rely on re-transmission of the information by the originating nodes. In this paper, we show that our proposed solution indeed performs better than the acknowledgment and erasure coding based solutions, and has the potential to effectively combat the high packet loss experienced by many underwater networks.

Keywords Network coding, erasure coding, underwater acoustic networks

1. Introduction

Although underwater networks are a form of wireless networks, they differ from typical radio wireless networks in a few significant ways [4]. The use of acoustics in underwater networks, rather than radio waves which are typically used in terrestrial wireless networks, results in significantly longer propagation delays and very limited data rates. The variability of the channel and high levels of ambient noise present in many ocean environments also leads to a high bit error rate (BER). Usually forward error correction (FEC) is used to add an appropriate level of redundancy and provide a low packet error rate (PER) at the expense of data rate. As the channel BER may not be known a priori and may vary during network operations, it is often difficult to choose an optimal FEC code to maximize the data rate while maintaining a low PER. While efforts are being made to address this problem by tuning the physical layer link parameters to optimize the effective data transfer rate [11], a simple solution that is often chosen is to select a FEC that provides an acceptable data rate for the application at hand. The packet loss (due to packet errors) then may be low if the channel is favorable, but may be high in other cases. Higher layer protocols are then expected to provide an appropriate level of reliability in the face of the packet loss through mechanisms such as ARQ, erasure coding, etc. In this paper we consider such a network with potentially high packet loss on some of the links, and propose a solution that efficiently transfers data across the network.

The approach to solving the problem described above depends on the network topology and the traffic pattern. We assume a fairly general scenario where all nodes are within acoustic transmission range of each other, and therefore may potentially be able to hear all other nodes in the network (with varying degree of packet loss). We also assume a uniform traffic demand i.e., all nodes in the network have data to transfer to all other nodes. The traffic may, however, either be unicast or multi/broadcast. We envision that this network model would be appropriate for many small distributed sensor networks, where all nodes generate data, and this data has to be distributed to other nodes in the network for processing and data fusion.

Reliable data transfer can be accomplished using unreliable links through the use of automatic repeat requests (ARQ) [10]. We explore this idea and show that a feasible, but inefficient, solution is obtained through the use of ARQ. We then explore an extension to rate-less erasure code based solutions for underwater point-to-point data

WUWNet'10 September 2010, Woods Hole, Massachusetts, USA.

transfer [3, 6] and show that they perform significantly better than the ARQ solution in some cases.

Network coding has become a very active area of research in wireless networks in the past decade [1]. Although the use of network coding has been explored in underwater networks [5, 9], the studies have mostly focused on multihop networks with a single source and sink. We finally explore the use of network coding to achieve effective data transfer over a fully connected but lossy network with uniform traffic demands. We show that a network coding solution is not only more efficient than an ARQ or erasure code based solution, but it is more robust to network link failures as it is able to implicitly utilize multihop routes in the network.

2. Problem Definition

Consider a lossy N-node wireless (acoustic) network with nodes indexed by $j \in \mathbb{Z}^+,\ j \leq N$. Let the probability of packet loss on a link from node i to node j be p_{ij}. We assume than any error in a packet leads to the packet being dropped and hence the probability of packet loss is equivalent to PER. The network is assumed to consist of a single *broadcast domain* i.e., a message transmitted by node i can be independently received by every node j (other than the transmitting node, $i \neq j$) with a probability $(1 - p_{ij})$. Each node has M messages to be transmitted. We denote the source messages originating at node j by $X_{jk},\ k \in \mathbb{Z}^+,\ k \leq M$. Each message X_{jk} is represented using an alphabet over the finite field $\mathbb{F}_q$ of size q.

We do not make any assumption on the destination of each message, but instead assume that the transfer is *completed* when all N nodes in the network are in the possession of all NM messages. We call this the *multicast model*. In the case of the ARQ-based solution outlined in section 3, we also consider a *unicast model*, where each message has a preassigned destination node, uniformly distributed over the set of all nodes (other than the source node).

Although for simplicity, we assume a TDMA MAC with N time slots per TDMA frame, the analysis and simulations presented here are not strongly dependent on this assumption and can easily be adapted to other MAC models. In the TDMA MAC, a time slot is assigned to each of the nodes for transmission. We index the TDMA frames by $t \in \mathbb{Z}$. The task is to formulate a solution that attempts to minimize the number of frames T required to complete the data transfer.

We expect that the transfer is completed in M frames in the absence of packet loss. With an average packet loss probability p, we expect that the transfer would take an average of $M/(1-p)$ frames at best. We compare the performance of various solutions by measuring the ratio of $M/(1-p)$ to the number of frames T required by the solution. We term this ratio as the *efficiency* η of the solution:

$$\eta = \frac{M}{(1-p)T} \tag{1}$$

where

$$p = \frac{1}{N(N-1)} \sum_{i,j\ \forall\ i \neq j} p_{ij} \tag{2}$$

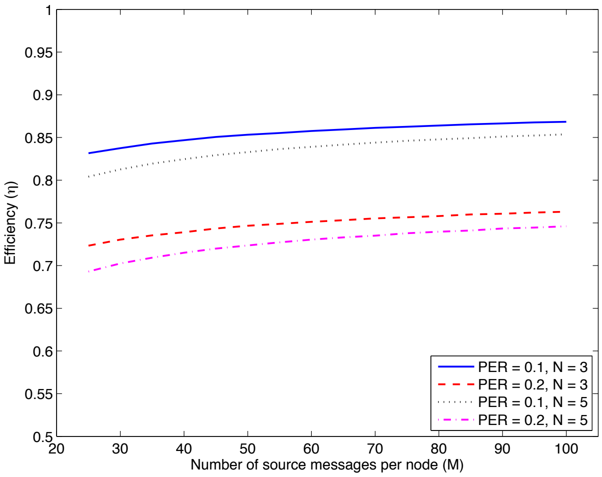

Figure 1. Performance of the ARQ solution in a small unicast network based on simulation.

3. ARQ Solution

A common solution to a reliable message delivery problem is via the use of ARQ [10]. In a ARQ solution, when a message remains unacknowledged, retransmissions ensure that the message is eventually reliably delivered.

In the ARQ solution considered in this paper, a node transmits a randomly chosen (or sequentially chosen) available source message during its assigned time slot within a frame. Depending on whether each message has a single or multiple destination nodes, the algorithm can be defined differently. In the unicast model, each message has a single intended destination node, while in the multicast model, each message must be successfully delivered to all other nodes. Regardless of which case is considered, a node's transmitted packet can piggyback a 1-bit acknowledgement for each of the messages sent by the other $N-1$ nodes in the previous $N-1$ time slots. For the unicast case, a node removes the source message from its list of available source messages for transmission once it has received the acknowledgement from that message's intended destination node. For the multicast case, a node can only do so upon receiving the acknowledgements from all the other $N-1$ nodes. For both cases, the data transfer is completed only when all nodes have exhausted all source messages.

Fig. 1 shows the efficiency of the unicast ARQ-based solution for a few small networks with varying PER and number of source messages. Fig. 2 shows similar results

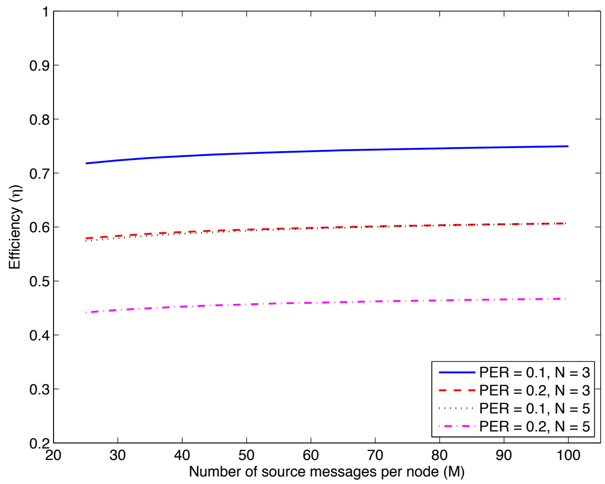

Figure 2. Performance of the ARQ solution in a small multicast network based on simulation. Note that the results for PER = 0.2, N = 3 and PER = 0.1, N = 5 are very similar and practically overlap.

of a multicast ARQ-based solution. The performance of this solution generally degrades with increasing PER and with increasing number of nodes, and improves slightly with increasing number of source messages. As one would expect, the performance of the multicast solution is poorer than the unicast solution since the multicast model requires that each message be successfully delivered to all other nodes in the network.

4. Erasure Coding Solution

Solutions based on FEC may be used as an alternative to the ARQ solution. A class of FEC codes known as erasure codes can be effectively used to combat packet loss in networks [8]. A set of source messages is expanded into a larger set of coded messages by an erasure code. These messages are transmitted across the network. The source messages can be recovered at each receiving node from a subset of the coded messages that are successfully received by that node. A successful decoding is only possible if the subset has certain properties (such as its size) that depend on the erasure code used. Solutions based on erasure coding have been previously proposed for underwater data transfer [3, 6]. In this section, we present a solution based on random rate-less erasure coding for a multicast network model.

4.1 Random Erasure Coding

In a solution based on random erasure coding, each node transmits a random linear combination (over $\mathbb{F}_q$) of its sources messages during each frame. Specifically, a message Y_{it} is transmitted by node i in frame t:

$$Y_{it} = \sum_{k=1}^{M} e_{ikt} X_{ik} \tag{3}$$

where $e_{ikt} \in \mathbb{F}_q$ are chosen randomly with a uniform distribution over $\mathbb{F}_q$. The message Y_{it} is received at every node j (other than the transmitting node, $i \neq j$) with probability $(1 - p_{ij})$. When a node j acquires M linearly independent messages from another node i, it is able to decode all M messages originating at node i. Writing the linear equations for the messages received at node j from node i in matrix form, we have:

$$\begin{pmatrix} Y_{i0} \\ Y_{i1} \\ \vdots \end{pmatrix} = \begin{bmatrix} e_{i10} & e_{i20} & \dots & e_{iM0} \\ e_{i11} & e_{i21} & \dots & e_{iM1} \\ \vdots & & \dots & \vdots \end{bmatrix} \begin{pmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{iM} \end{pmatrix} \tag{4}$$

with only the messages Y_{it} that were successfully received at node j participating on the left hand side, and the corresponding e_{ikt} participating on the right hand side. When at least M linearly independent messages are received, the matrix on the right hand side is full rank, and (4) can be solved for the source messages $X_{ik}\ \forall\ k$. Efficient techniques for solving such equations exist [2, 7]. When all nodes are able to decode all messages from all other nodes, the data transfer is completed.

At the end of the data transfer, all N nodes are in possession of all NM messages. This meets the multicast model's requirements. However, since the unicast model's requirements are also met by this, the solution may be used in either model.

4.2 Performance Analysis

Let $n_{ij,t}$ be the number of linearly independent messages received by node j from node i at the end of frame t. Since this number is a random variable, we model its distribution with $x_{ij,t,n}$, the probability that a node j has received n independent messages from node i by the end of frame t. At the end of the first frame ($t = 0$), we have:

$$\begin{aligned} x_{ij,0,0} &= p_{ij} \\ x_{ij,0,1} &= 1 - p_{ij} \\ x_{ij,0,n} &= 0 \quad \forall\, 2 \leq n \leq M \end{aligned} \tag{5}$$

At the end of any frame $t \geq 1$, we have:

$$\begin{aligned} x_{ij,t,0} &= x_{ij,t-1,0}\, p_{ij} \\ x_{ij,t,n} &= x_{ij,t-1,n}\,[p_{ij} + (1 - p_{ij})\phi_{n,M}] \quad \forall\, 1 \leq n \leq M \\ &+ x_{ij,t-1,n-1}(1 - p_{ij})(1 - \phi_{n-1,M}) \end{aligned} \tag{6}$$

where

$$\begin{aligned} \phi_{n,M} &= \frac{q^n - 1}{q^M - 1} \quad \text{if } M > 0 \\ \phi_{n,0} &= 1 \end{aligned} \tag{7}$$

is the probability that a transmitted message randomly chosen over a M-dimensional space lies in the sub-space spanned by the n linearly independent messages already known at the receiver (we assume that an all-zero message is never transmitted). The above expression can be written in a matrix form:

$$\begin{aligned} \mathbf{x}_{ij,t} &= \mathbf{A}_{ij}\mathbf{x}_{ij,t-1} \quad \forall\, t \geq 1 && (8) \\ &= \mathbf{A}_{ij}^{t}\mathbf{x}_{ij,0} && (9) \end{aligned}$$

where $\mathbf{A}_{ij}$ is a $(M+1)\times(M+1)$ matrix with non-zero entries given by

$$\begin{aligned} A_{ij,0,0} &= p_{ij} && (10) \\ A_{ij,n,n} &= p_{ij} + (1-p_{ij})\phi_{n,M} && \forall\, 1 \leq n \leq M \\ A_{ij,n+1,n} &= 1 - A_{ij,n,n} && \forall\, 0 \leq n < M \end{aligned}$$

The expected number of linearly independent messages known at the receiver at the end of frame t is given by

$$\bar{n}_{ij,t} = \mathbb{E}[n_{ij,t}] = \sum_{n=0}^{M} n x_{ij,t,n} \tag{11}$$

and the probability that all the message from node i are known at node j by the end of frame t is $x_{ij,t,M}$. The expected number of frames required to reach state M where all messages are known at the recipient is given by

$$\begin{aligned} \bar{T}_{ij}^{\text{ec}} &= \mathbb{E}[\min t+1 | n_{ij,t} = M] && (12) \\ &= \sum_{t=M-1}^{\infty} (t+1)(x_{ij,t,M} - x_{ij,t-1,M}) \end{aligned}$$

For a network with N nodes, we have $N(N-1)$ such independent logical links with M messages to be transmitted over each link. The expected number of frames to completion of transfer for all nodes is given by

$$\bar{T}^{\text{ec}} = \sum_{t=M-1}^{\infty} (t+1) \left(\prod_{i,j} x_{ij,t,M} - \prod_{i,j} x_{ij,t-1,M} \right) \tag{13}$$

where the product terms are for all $i \neq j$. If $p_{ij} = 1$ for any $i \neq j$ then the probabilities $x_{ij,t,n}$ do not change with t. In this special case, (12) and (13) cannot be used. We instead have $\bar{T}^{\text{ec}} = \bar{T}_{ij}^{\text{ec}} = \infty$ as the data transfer can never be completed.

The computed efficiencies corresponding to (13) for binary erasure codes ($q = 2$) and various values of PER and N are shown in Fig. 3. These analytical results are compared with the equivalent estimates from simulations. We can see that there is a fair agreement between analysis and simulation.

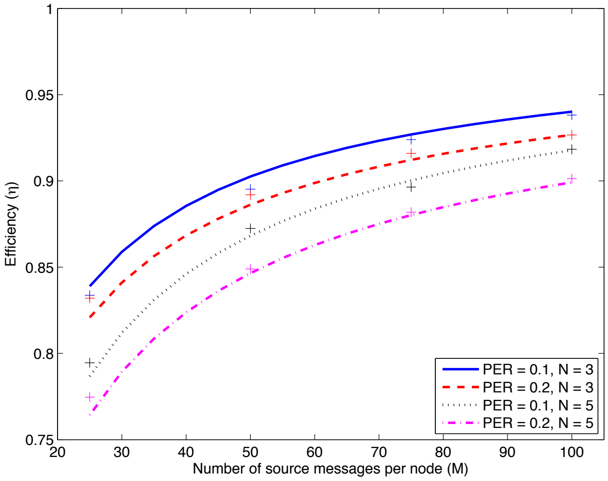

Figure 3. The predicted performance of binary erasure coding from analysis fairly agrees with simulations. The lines show the analytical results, while the "+" markers show the corresponding simulation results.

5. Network Coding Solution

The erasure coding solution in the previous section sends messages composed from multiple source messages originating at that node. Inspired by the idea of network coding [1], we extend the combining of messages to include not only the source messages originating at a node, but also the composite messages successfully received from other nodes.

5.1 Random Network Coding

In the random network coding solution, each node transmits a random linear combination (over $\mathbb{F}_q$) of its sources messages and messages received by the node. If $\hat{Y}_{jt} \subset \{Y_{i\tau} \;\forall\; i \neq j,\; \tau < t\}$ is the set of messages received at node j by the start of frame t, then:

$$Y_{it} = \sum_{k=1}^{M} e_{ikt} X_{ik} + \sum_{h=1}^{|\hat{Y}_{it}|} \epsilon_{iht} \hat{Y}_{ih} \tag{14}$$

where e_{ikt} and $\epsilon_{iht} \in \mathbb{F}_q$ are chosen randomly with a uniform distribution over $\mathbb{F}_q$. Since the messages in $\hat{Y}_{jt}$ are also linear combinations of source messages X_{ik}, the resulting message can be expressed as a linear combination of all source messages in the network:

$$Y_{it} = \sum_{j=1}^{N} \sum_{k=1}^{M} \hat{e}_{ijkt} X_{jk} \tag{15}$$

where $\hat{e}_{ijkt}$ are fully determined by the coefficients e_{ikt} and ϵ_{iht} chosen in (14). The transmitted message Y_{it} is received at every node j (other than the transmitting node, $i \neq j$) with probability $(1-p_{ij})$. When a node j acquires

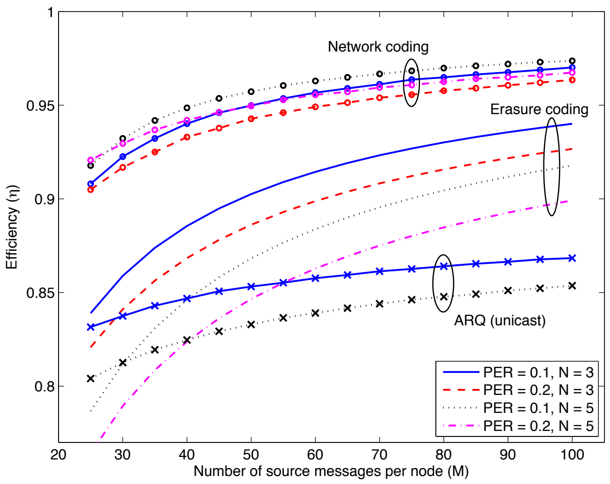

Figure 4. The performance of binary random network coding (lines with circles) is significantly better than that of binary random erasure coding (lines without markers) and of the ARQ unicast solution (lines with crosses). Unlike erasure coding and the ARQ solution, the network coding performance marginally improves with increased number of nodes and is only weakly dependent on PER. Note that the ARQ multicast performance for all cases, and the unicast performance of two of the four cases is not visible in this graph as the efficiency is lower than 0.78 (the minimum value on the y-axis).

NM linearly independent messages from all nodes, it is able to decode all NM messages in the network. Writing the linear equations for all messages received at node j in matrix form, we have:

$$\begin{pmatrix} Y_{00} \\ Y_{01} \\ \vdots \\ Y_{10} \\ \vdots \end{pmatrix} = \begin{bmatrix} \hat{e}_{0110} & \hat{e}_{0120} & \cdots & \hat{e}_{0NM0} \\ \hat{e}_{0111} & \hat{e}_{0121} & \cdots & \hat{e}_{0NM1} \\ \vdots & & \cdots & \vdots \\ \hat{e}_{1110} & \hat{e}_{1120} & \cdots & \hat{e}_{1NM0} \\ \vdots & & \cdots & \vdots \end{bmatrix} \begin{pmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{NM} \end{pmatrix} \tag{16}$$

with only the messages Y_{it} that were successfully received at node j participating on the left hand side, and the corresponding $\hat{e}_{ijkt}$ participating on the right hand side. When at least NM linearly independent messages are received, the matrix on the right hand side is full rank, and (16) can be solved for the source messages $X_{jk}\ \forall\ j, k$. Efficient techniques for solving such equations exist [2, 7]. When all nodes are able to decode all the messages, the data transfer is completed. As in the case of erasure coding, although this solution directly applies to the multicast model, it also meets the requirements of the unicast model.

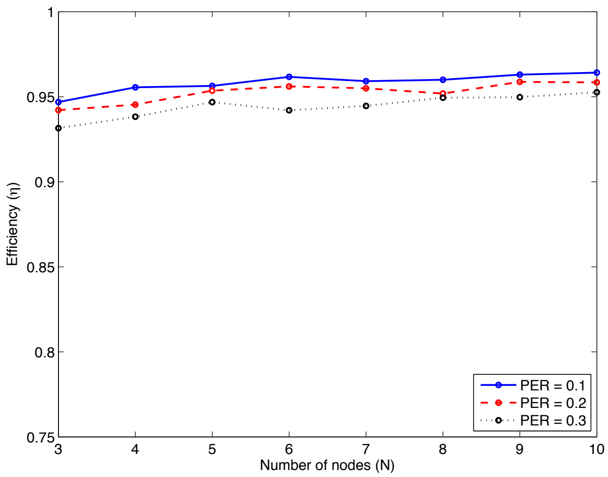

Figure 5. The performance of network coding is not very sensitive to the number of nodes and only weakens marginally with increasing PER. The results shown here are simulated for binary random network codes with $M = 50$.

5.2 Performance Comparison

Fig. 4 shows the performance of a binary ($q = 2$) random network coding solution as compared to the ARQ and erasure coding solutions. As analytical results for the ARQ and network coding solutions are not available, the performance results for these cases were obtained via simulation. The network coding solution is able to complete the data transfer with a much higher efficiency as compared to the ARQ or erasure coding solutions. It is noteworthy that although the network coding solution is a multicast solution, it outperforms even the unicast ARQ solution and can be used even if the requirement is only to support unicast. The performance of network coding only weakly depends on the PER. The performance is also only weakly dependent on the number of nodes in the network, with a larger number of nodes giving marginally better performance. Contrast this to the case of ARQ and erasure coding solutions where the performance degrades with increasing number of nodes in the network and with increasing PER.

Fig. 5 shows the performance of the binary random network coding solution for networks of varying sizes (number of nodes) and varying PER. We can see that the performance is fairly insensitive to the size of the network and hence the solution is scalable. The performance only degrades very slowly with increasing PER, and hence may be used in networks with high PER.

5.3 Implicit Routing using Network Coding

In the network coding solution described above, the information conveyed by a particular source message may travel along different routes with varying number of hops before

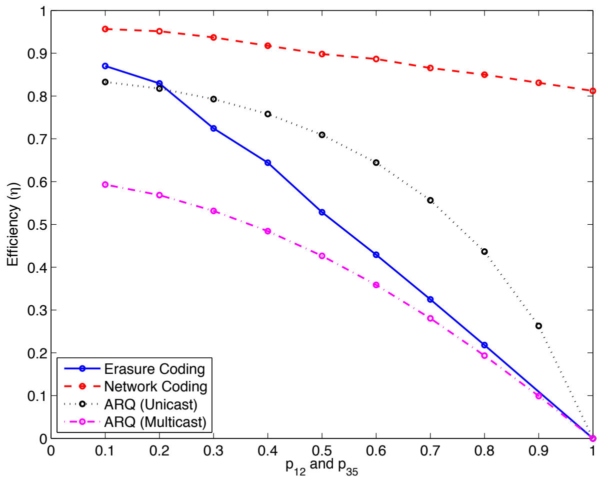

Figure 6. Simulation results ($M = 50$, $q = 2$) for a 5-node network with PER = 0.1 on all links except links 1→2 and 3→5. As the PER on those two links increases, we see that the network coding solution still performs well whereas the ARQ and erasure coding solutions rapidly degrade in performance.

reaching a given destination node. Although the network model we use has a single broadcast domain, the network coding solution effectively routes the information over the network and is therefore insensitive to a few poor links or link failures.

The information only travels along the direct link between each source-destination node pair in the ARQ and erasure coding solutions. No explicit routing or relaying protocol is implemented in our analysis or simulations. Since we require that all nodes receive the source messages from all other nodes for the data transfer to be considered complete, a link failure (PER = 1) causes the data transfer to never complete in the case of ARQ or erasure codes. However, as we shall next demonstrate, the failure or a small fraction of the links in a network does not have such a strong effect on the solution with network coding.

So far we have only shown results for networks with the same PER on all links in the network. We now turn our attention to networks with a few poor (or failed) links to demonstrate the insensitivity of network coding to link failures. We simulate a 5-node network with $p_{ij} = 0.1 \ \forall \ i \neq j$, $q = 2$ and $M = 50$. We progressively degrade two randomly chosen links (link 1→2 and link 3→5 with PERs p_{12} and p_{35} respectively) in the network to increase the PER to 1 and study the effect on the performance of the various solutions. The results are shown in Fig. 6. As expected, the network coding solution performs well even when the PER on the two poor links is very high, whereas the ARQ and erasure coding solutions rapidly degrade in performance as the links progressively fail.

6. Conclusions

In this paper, we considered the problem of transmitting data efficiently in a small wireless lossy network with uniform traffic demand. We studied solutions based on ARQ, erasure coding and network coding, and showed that the network coding based solution was able to transfer data more efficiently than all the other solutions under consideration. Efficiencies as high as 90-95% could easily be achieved and the performance was found to be relatively insensitive to the size of the network, the packet loss probability and the amount of data to be transferred. A unique property of the network coding solution is its ability to use implicit routing to achieve a high degree of robustness to isolated link failure. This property may be critical to many underwater networks where link quality can vary significantly depending on the channel conditions. We conclude that network coding provides a very good solution to the problem of data transfer in a small network with high packet loss.

References

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, 2000.

[2] V. Bioglio, M. Grangetto, R. Gaeta, and M. Sereno. On the fly gaussian elimination for LT codes. *Communications Letters, IEEE*, 13(12):953 –955, December 2009.

[3] M. Chitre and M. Motani. On the use of rate-less codes in underwater acoustic file transfers. In *OCEANS 2007 - Europe*, June 2007.

[4] M. Chitre, S. Shahabudeen, and M. Stojanovic. Underwater acoustic communications and networking: Recent advances and future challenges. *The Spring 2008 MTS Journal, "The State of Technology in 2008"*, 42(1):103–116, 2008.

[5] Z. Guo, B. Wang, P. Xie, W. Zeng, and J.-H. Cui. Efficient error recovery with network coding in underwater sensor networks. *Ad Hoc Netw.*, 7(4):791–802, 2009.

[6] O. Kebkal. On the use of interwoven order of oncoming packets for reliable underwater acoustic data transfer. In *OCEANS 2009-EUROPE, 2009. OCEANS '09*, pages 1–7, May 2009.

[7] S. Kim, K. Ko, and S.-Y. Chung. Incremental Gaussian elimination decoding of raptor codes over BEC. *Communications Letters, IEEE*, 12(4):307 –309, April 2008.

[8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2):569–584, 2001.

[9] D. E. Lucani, M. Medard, and M. Stojanovic. Network coding schemes for underwater networks: the benefits of implicit acknowledgement. In *Proceedings of the second workshop on Underwater networks*. ACM, 2007.

[10] L. Peterson and B. Davie. *Computer networks: a systems approach*. Morgan Kaufmann Pub, 2007.

[11] S. Shankar, M. Chitre, and M. Jayasuriya. Data driven algorithms to tune physical layer parameters of an underwater communication link. In *OCEANS'10 Sydney*. IEEE, May 2010.