Reliable Broadcast with Joint Forward Error Correction and Erasure Codes in Wireless Communication Networks

Pouya Ostovari and Jie Wu
Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
Email: {ostovari, jiewu}@temple.edu

Abstract—**One of the main challenges in wireless networks is addressing the unreliability of the wireless links, and providing reliable transmissions. Two important sources of errors in wireless transmissions are noise and interference. In order to address the errors due to noise, forward error correction methods can be uses, in which redundancy is added to the packets to detect and correct the bit errors. However, when the environment is too noisy, or there is interference among the transmissions, forward error correction codes might not be able to correct the bit errors, which results in packet erasures. In this case, application-layer erasure code, such as network coding, are useful. In this paper, we consider a wireless network which faces both random bit errors and packet erasures. In order to provide reliable transmissions, we benefit from joint forward error correction and erasure codes. We formulate the problem as an optimization problem. We also propose a low-complexity method to find the optimal redundancy that should be assign to the forward error correction and erasure code. Our method consists of two-phases: offline and online phases. In the offline phase, we generate a reference table, which shows the successful delivery of the packets for each possible transmission strategy. The source node uses this reference table in the second phase to find the optimal strategy depending on the noise and interference level. We show the effectiveness of our proposed method through extensive simulation.**

Index Terms—**Broadcast, wireless networks, packet erasure, random linear network coding, forward error correction, reed-solomon, error-prone channel.**

I. Introduction

In the recent years, the hardware technology of the mobile devices, e.g. smartphones and tablets, is developing very fast. These devices are becoming very popular and a part of our daily life. The wireless devices are used for a variety of purposes, including but not limited to surfing the Internet, listening to online music, and video streaming. These devices typically use cellular or WiFi connections to provide the Internet connection. However, a main challenge in wireless networks is the addressing the unreliability of the wireless links, and providing reliable and robust transmissions.

Providing reliable wireless transmissions has been widely studied by the research community. The automatic repeat request (ARQ) [1] method is the easiest and most frequently used technique to provide reliable transmissions. In ARQ, the receiver nodes send a feedback message after each set of transmissions to report the received or lost packets. However, feedback messages incur transmission overhead. Another widely used approach for reliable transmission is to use forward error correction (FEC) codes [2]–[4], in which redundancy is added to the packets to detect and correct the bit errors. In hybrid-ARQ methods, FEC and ARQ messages are combined together in order to reduce the overhead of the ARQ method [5], [6].

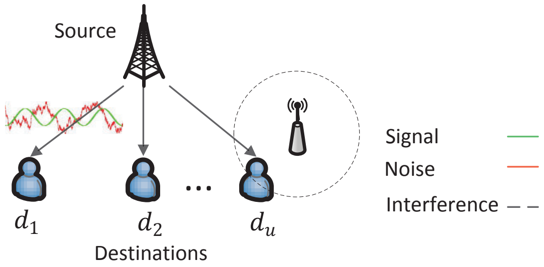


Fig. 1. Motivation

As Figure 1 shows, noise and interference are two major sources of transmission errors in wireless networks. The received signals by the receiver nodes are never exactly the same as the transmitted signals by the source node. This is the result of the noise that exist in the wireless environment. The physical layer can remove the effect of the noise if the noise level is less than a specific threshold. However, in the case that the strength of the noise is greater than the threshold, the physical layer cannot correct the signal and distinguish the correct bits. In order to detect and refine the bit errors, FEC methods can be used [2]. Reed-Solomon code [7] is an example of a FEC code.

Interference among the nodes is the second major origin of errors in wireless transmissions. In contrast with the wired links, wireless medium is shared among the wireless devices. Consequently, in the case that two wireless transmitter that have a common node in their communication rage transmit simultaneously, the receiver node cannot receive the transmitted packet correctly. Feedback messages can be used to report the lost packets. However, feedback messages put a burden on the system, specifically in the multicasting applications. Because of this overhead, many multicat applications avoid using feedback messages. In order to make the wireless transmissions robust against interference and the other sources of packet erasures, erasure codes [8] can be used. Using erasure codes, a set of m packets are encoded to $M > m$ packets, such that receiving a subset of the M encoded packets is sufficient to decode and retrieve the original m packets. Random linear network coding (RLNC) [9], [10] and fountain [11]–[13] codes (also known as rateless codes) are two well-known examples of erasure codes.

The authors in [14] performed an experiment in wireless sensor networks to study the advantage of joint FEC and erasure coding. The authors show the effectiveness of using random linear network coding [9] as an erasure code in the hostile wireless environment that the sensor networks work in. They show that random linear network coding can outperform the physical-layer FEC in the case of high packet erasure rate. However, the authors did not propose any method to find the optimal transmission strategy.

In this work, we study the problem of robust data transmission in wireless networks. We consider a set of packets that should be broadcast to a set of set of wireless users. In order to make the transmitted packets robust against noise and interference, we use joint FEC and erasure codes, which are intra- and inter-packet redundancy, respectively. Our goal in this papers is to use a given number of transmissions and assign the available redundancy to FEC and erasure codes in way that maximizes the probability of successfully delivering the set of original packets. We formulate the problem as an optimization problem. Moreover, we propose a two-phases algorithm to find the optimal strategy. In the first phase of the algorithm, we create a reference table, which shows the success probability of each transmission strategy for every noise and interference probability. This phase is performed offline and only once. In the second phase, the source node searches the reference table to find the optimal distribution of the redundancy among the FEC and erasure code.

The remainder of this paper is organized as follows. In Section II, a background on network coding is provided and the related work is reviewed. We introduce the setting and the problem definition in Section III. We propose our joint FEC and erasure coding method in Section IV. Finally, Section VI concludes the paper.

II. Related Work and Background

A. Reliable Transmission

A simple way to address error-prone wireless transmissions and to provide reliable transmissions is to use feedback messages. The most frequently used mechanism to ensure successful delivery of the packets is the automatic repeat request (ARQ) method [1]. However, the ARQ method puts some transmission overhead on the network, which is the result of transmitting feedback messages. This overhead becomes a major challenge in multicast applications, such that many multicast applications do not use ARQ. In order to reduce the overhead of the ARQ method, hybrid-ARQ approaches [6], [15] are proposed. In hybrid-ARQ, FEC is combined with the ARQ. The used FEC methods in the hybrid-ARQ schemes increases the success delivery rate of the transmitted packets, which reduces the required number of feedback messages. The RMDP method uses Vandermonde [16] code combined with ARQ to ensure reliability.

Using erasure codes [8] is an efficient way to provide reliable transmissions without need for feedback messages. In erasure codes, a set of m packets are encoded to $M > m$ coded packets. A destination is able to decode the coded packets once it receives a sufficient number of coded packets. Fountain (rateless) codes [11], [12], such as LT and Raptor codes, are examples of erasure codes. Using fountain codes, a source can potentially generate an unlimited number of coded packets. The source can transmit the packets until the destination nodes receive a sufficient number of coded packets and decoded the coded packets. The benefit of using fountain codes is that, the coded packets contribute the same amount of information to the destination nodes, and the destination node only needs to receive enough coded packets, regardless of which packets has been received or lost. Typically XOR operation is used in fountain codes to code the packets.

In order to combat with the bit errors during wireless transmissions, forward error correction (FEC) codes [2]–[4] are widely used at the physical layer. A FEC code converts a packet of n bits to a packet of $n + k_b$ bits. FEC codes can detect and correct at most k_b bit errors. The rate of a FEC code is defined as the ratio of the original packet size to the packet size after adding the FEC bits. The works in [17], [18] studied the problem of designing efficient FEC codes.

In order to evaluate the synergy between FEC and erasure codes, the authors in [14] perform an experiment in a low power wireless sensor. They use random linear network coding as an erasure code, and combine it with physical-layer FEC. The motivation of the authors was the hostile wireless environment that low power sensors operate in, in which there is a significant interference from nearby nodes. The results of the paper show the effectiveness of random linear network coding, outperforming physical-layer FEC, in the case of high packet erasure rate. However, the authors do not propose any method to find the optimal transmission strategy.

B. Network Coding

Network coding (NC) [10], [19]–[22] is proposed in [23] for the first time. The authors in [23] use network coding to solve the bottleneck problem in a single multicast problem. It is shown that network coding enables us to achieve min-cut max-flow in the case of multicast application in wired networks. The authors in [24] depict that linear network coding achieves the capacity of a single multicast problem. The idea of random linear network coding is proposed in [9]. The authors show that if we select the coefficients of the linearly coded packets randomly, with a high probability all of the coded packets will be linearly independent. The authors in [25] derive a useful algebraic model of the linear network coding.

The main idea in random linear network coding is to use random coefficients to code the packets together. The operations are performed on a finite fields (Galois field). In random linear network coding, each packet has a form of $\sum_{j=1}^{m} \alpha_j \times P_j$. Here, we represent the original packets as

P_j. Moreover, α_j is a random coefficient. Similar to the fountain codes, a source node can use random linear network coding to generate and transmit infinite number of coded packets. A destination node is able to decode the linearly coded packets and retrieve the original packets if it has access to m linearly independent coded packets. The decoding is similar to solving a system of linear equations. For this purpose, Gaussian elimination can be used. In the applications that 100% reliability is needed, the source node can keep transmitting random linearly coded packets until it receives a single acknowledgment message from the destination node.

Reliable one-hop multicasting transmission has been studied in [26]–[28]. In order to provide 100% reliability, the authors use feedback messages. Using these feedback messages, the source node can find the lost packets and retransmit them. In the proposed methods in [26]–[28], the source node benefit from XOR coding in the retransmissions. In this way, the number of required transmissions to deliver to lost packets decreases, which improves the system throughput. The idea is to combine the packet that is lost by a destination node, but has been received by the other destination node. Using network coding in the retransmission phase, each transmitted coded packet can deliver multiple lost packets to the different destination nodes.

The concept of symbol-level network coding is proposed in [29]. The authors show that, the throughput of symbol-level network coding is more than that of the packet-level network coding. The idea is that, even in the case of erroneous transmissions, some parts of the transmitted packets might be received correctly ad without any error. As a result, if we can distinguish and use these parts of the transmitted packets, we can save the transmissions. Consequently, the throughout of the system increases and the transmission cost reduces.

In [30], [31], the concept of symbol-level network coding is extended to the application of distributing information in vehicular networks. The authors depict that symbol-level network coding is an efficient way to designate data in vehicular ad-hoc networks. In [31], symbol-level network coding is used to efficiently distribute multimedia data in vehicular ad-hoc networks.

III. System Setting

We consider a single-hop wireless network that consists of one source and u destination nodes d_i, as depicted in Figure 2. The source node, which can be a LTE base station or a WiFi router, has a batch of packets to send to the destination nodes. The packets are segmented into sets of m packets. Also, the size of each packet is equal to n bits. In the rest of the paper, we use destination and users exchangeably.

We consider two sources of errors that might affect the packet transmissions. The first source of error is the noise in the environment, which results in random errors in the transmitted bits. The second source is interference among the nodes, which results in burst errors in the bits, and might issue packet erasure. We represent the probability of bit errors at destination i as $\epsilon_{i,b}$. Moreover, the interference probability at node i is represented as $\epsilon_{i,I}$.

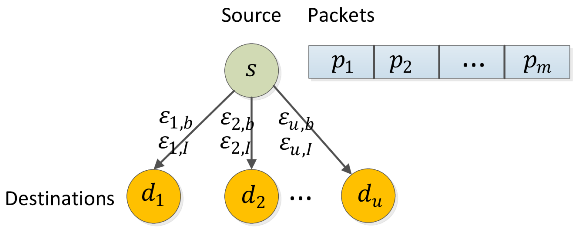


Fig. 2. System setting

The source node can totaly send X bits for any set of m packets. The limited number of bit transmissions can be motivated by real-time applications or delay sensitive data, such as video streaming. The redundant bits can be assigned as intra-packet (bit level) forward error correction (FEC) to battle the random bit errors. They can also be assigned as the application-level erasure codes (or inter-packet redundancy) to provide protection against packet erasures. We represent the amount of added FEC to each packet as k_b. Moreover, the number of redundant transmitted packets is shown as k_p. Our objective in this work is to provide protection for the transmitted packets, such that the probability of receiving the batch of m packets by the destination nodes is maximized. In other words, we want to find the optimal transmission scheme that maximizes the probability of successfully receiving the set of m packets in each segment. Table I summarizes the set of notations used in this paper.

IV. Robust Transmission Scheme

Because of more sources of errors and the shared nature of wireless medium, wireless links are less reliable than the wired links. Noise and interference are two major sources of transmissions errors in wireless networks. Noise always exists in the wireless environments, and it changes the transmitted signals. In the case that noise level is greater that a threshold, the physical layer cannot correct the received signals to distinguish the actual bit. In order to correct the bit errors, forward error correction (FEC) methods [2]–[4] are used. For example, in order to find a single bit error, a parity bit can be added to the packet. Parity can be the logical XOR (exclusive disjunction) of the bits. In order to find more number of errors and correct the errors, more complex FEC methods can be used. Using FEC methods, a packet of n bits information is mapped to a $n + k_b$ bit packet, which results in a lower error rate. In this case, the effective rate, also called coding rate, is equal to $n/(n + k_b)$. This type of redundancy is called inner or intra-packet redundancy.

The next major source of errors in wireless transmissions is interference among the wireless nodes. Wireless medium is a shared medium. As a result, once two nodes that has a common node in their communication range transmit at the same time, they will interfere with each other. In order to provide reliable transmissions, feedback messages can be used to find the lost

TABLE I
THE SET OF SYMBOLS USED IN THIS PAPER.

Notation	Definition
$\epsilon_{i,b}$	The bit error rate at user d_i
$\epsilon_{i,e}$	The erasure probability due to noise at user d_i
$\epsilon_{i,I}$	The interference probability at user d_i
ϵ_i	The packet erasure probability due to noise and interference at user d_i
$\epsilon_{i,s}$	The successful decoding probability at user d_i
m	Number of packets to send
n	Packet size (before adding FEC)
k_b	Number of FEC bits
k_p	Number of redundant packets
g	The step size for amount of FEC
u	Number of destination nodes
d_i	The i-th destination node

packets. However, feedback messages might not be feasible. They also incur some overhead and transmissions delay, since the transmitter needs to stop transmissions to receive feedback messages. Specifically, in multicast or broadcast applications, feedback messages are very costly. In order to improve the communication reliability without using feedback messages, erasure codes [8] can be used. In erasure coding, cross-packet (inter-packet) redundancy is added to the packets. For example, m packets are encoded to $m + k_b$ packets, where k_b is the number of redundant encoded packets. In this case, the coding rate (or effective rate) is equal to $m/(m + k_b)$. An erasure code is optimal if the original m packets can be retrieved using any subset of m out of the $m + k_b$ encoded packets. Random linear network coding (RLNC) [9], [10] is such a coding technique. Fountain codes are also effective erasure codes that can be applied to provide reliable transmissions without need for feedback messages; however, fountain codes are not optimal erasure codes.

In the following sections, we first discuss about the FEC and erasure codes that we use in our work. We then calculate the probability of successful packet reception for a given FEC and erasure code redundancy. At the end, we propose a two-phases method to find the optimal redundancy distribution to the FEC and erasure coding.

A. Forward Error Correction

At the packet level, we use forward error correction to battle with the random bit errors, which might happen due to the environment noises. For this purpose, we choose Reed-Solomon (RS) code. Reed-Solomon code is a typical FEC code, which has been widely studied and applied against for error correction. In RS$(n, n+k_b)$, for every n bits of a packet, k_b parity bits are added to the packet. If a destination node receives at least n out of the $n+k_b$ bits correctly, it will be able to retrieve the original packet (the n bits). Reed-Solomon code is an optimal code, since receiving any n bits is sufficient to retrieve the original n bits. It should be noted that other FEC methods can also be applied in our method.

B. Application Layer Erasure Code

In order to address packets erasures that might be due to interference or high bit error rate, erasure codes can be used. Random linear network coding and fountain codes are two widely used erasure codes. The complexity of coding and decoding in fountain codes is less than that of the random linear network coding. However, in contrast with random linear network coding, fountain codes are not optimal.

If fountain code is used to code m packets, $(1 + \beta)m$ [11] coded packets are required to decode and retrieve the original m packets. Here, β is a small number, which is the overhead of the fountain code. It should be noted that $(1 + \beta)m$ is the required number of received coded packets for decoding, and is independent of the channel reliability. It is shown that this overhead goes to zero as m goes to infinity [13]. In our model, the number of packets that should be transmitted are limited. As a result, because of this overhead, we prefer to use random linear network coding in our proposed method.

C. Joint Network Coding and FEC Scheme

As mentioned above, FEC and random linear network coding can be used to address the bit errors and erasures in wireless networks. As a result, to address bit errors and erasures that might happen during the transmissions of a set of packets, in our robust transmission method, we use both of these schemes.

Figure 3 shows the overview of our coding scheme. We first partition the packets that need to be transmitted to a set of equal size segments. The size of each segment is equal to m packets. The reason for segmentation is to reduce the encoding and decoding complexity of the random linear coded packets. Moreover, segmentation reduces the decoding delay of the packets. Without segmentation, a receiver cannot use any received coded packet until it receives a sufficient number of coded packets. Then, the receiver can use Gaussian elimination to decode and retrieve all of the original packets. In the case of transmitting real-time data, such as video streaming, the receiver node needs to decode the coded packets and retrieve the original packets in a timely fashion.

After performing segmentations, the packets of each segment are linearly coded together using random coefficients. In this way, all of the coded packets have the same importance and contribute the same amount of data to the destination nodes. The number of packets in each segment is m, and the number of generated coded packets is $m + k_p$, where k_p is the number of redundant transmitted packets. This coding is an inter-packet coding and makes the transmission method robust against packet erasures. The coded packets are shown in Figure 3. This process is performed at the applications layer.

In order to make each packet transmission robust against the noises and reduce the packet erasure probability, we apply a FEC method on each coded packet. After performing random linear network coding, the physical layer receives the linearly

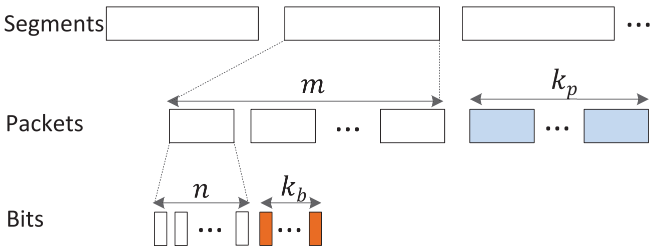


Fig. 3. Network coding and FEC scheme

coded packets and performs the Reed-Solomon scheme on the coded packets. The size of each coded packet before and after adding the FEC redundancy is equal to n and $n + k_b$. Here, we represent the added intra-packet redundancy (FEC) to each coded packet as k_b.

The question that arises is that, what is the best scheme to distribute the redundancy among the inter-packet (random linear network coding) and the intra-packet (FEC) coding. It is clear that in the case of a high bit error rate, more redundancy should be assigned to intra-packet coding. In contrast, in order to make the transmissions with high erasure probability robust, more redundancy should be assigned to the inter-packet erasure coding. However, finding a redundancy distribution that maximizes the probability of successfully receiving the packets is not straightforward. In the next two sections, we first formulate the problem and calculate the probability of successfully receiving the packets by a destination node. We then propose an algorithm to find the optimal solution.

D. Formulation

For each set of packet to be transmitted, the total number of bits that can be transmitted should be less than or equal to X. Therefore, we have $(n+k_b) \times (m+k_p) \leq X$. We represent the error probability at bit level, interference probability, erasure provability due to noise, and packet erasure probability that user d_i experiences as $\epsilon_{i,b}$, $\epsilon_{i,I}$, $\epsilon_{i,e}$, and ϵ_i, respectively. We can calculate the packet erasure probability due to noise (bit level errors) as:

$$\epsilon_{i,e} = 1 - \sum_{j=n}^{n+k_b} \binom{n+k_b}{j} \epsilon_{i,b}^{n+k_b-j} (1-\epsilon_{i,b})^j \tag{1}$$

where, n and k_b are the packet size and the number of FEC bits. Reed-Solomon code that is used as the FEC scheme can detect and correct up to k_b errors. In the above equation, we calculate the probability of correctly receiving at least n bits, and subtract if from 1. If the number of errors are greater than k_b, the bit errors cannot be corrected, and the packet is discarded. In the case that noise does not cause packet erasure, interference can still result in packet erasure. As a result, the total erasure probability can be calculate as follows:

$$\epsilon_i = \epsilon_{i,e} + (1-\epsilon_{i,e}) \times \epsilon_{i,I} \tag{2}$$

Assume that the number of data packets to send is m. Also, we represent the number of redundant transmitted packets as k_p. A destination node is able to decode the linearly coded packets and retrieve the original m packets if it receive m linearly independent packets. It is shown in [9] that with a high probability, any set of m linearly coded packets are linearly independent. As a result, with a high probability, receiving any m coded packets is sufficient for decoding. We represent the probability of successfully receiving the set of m packets by user d_i as $\epsilon_{i,s}$. We can calculate the success decoding probability as follows:

$$\epsilon_{i,s} = \sum_{j=m}^{m+k_p} \binom{m+k_p}{j} {\epsilon_i}^{m+k_p-j} (1-\epsilon_i)^j \tag{3}$$

If we define the objective function as the summation of successfully receiving packets probability of all the destinations, the optimization problem can be summarized as follows:

$$\begin{aligned} & \max \sum_{i=1}^{u} \epsilon_{i,s} \\ s.t \quad & \epsilon_{i,e} = 1 - \sum_{j=n}^{n+k_b} \binom{n+k_b}{j} \epsilon_{i,b}^{n+k_b-j} (1-\epsilon_{i,b})^j, \\ & \forall i: \; 1 \leq i \leq u \\ & \epsilon_i = \epsilon_{i,e} + (1-\epsilon_{i,e}) \times \epsilon_{i,I}, \quad \forall i: \; 1 \leq i \leq u \\ & (n+k_b) \times (m+k_p) \leq X \end{aligned}$$

E. Finding Optimal Distribution

Unfortunately, Equations (1) and (3) cannot be simplified in a closed form. As a result, the complexity of the proposed optimization in the previous subsection is high. In order to reduce the complexity of finding the optimal FEC and random linear network coding redundancy levels, we propose a two-phases scheme, which contains offline and online phases. In the offline phase, a reference table is created which shows the success delivery of the packets for each possible FEC and random linear network coding redundancy levels. Then, in the second phase, the transmitter performs a fast search algorithm on the reference table in order to find the optimal FEC and random linear network coding levels depending on the noise and interference probabilities of the users. In the following subsections, we discuss about these two-phases.

1) Reference Table Creation: The first phase of our method is creating a reference (look-up) table, which for each possible distribution of the transmissions and error probability shows the probability of successfully decoding and retrieving the original m packets by a destination node. It should be noted that this phase needs to be run only once to generate a reference table, and the source node can use the created reference table to find the optimal distribution of the transmissions in any error rate (random bit error and interference probability). Using this reference table, the transmitter can perform a simple search to find the optimal transmission scheme for any number of destination nodes and error rate.

In order to create the reference table, we generate all of the possible distributions of the X transmissions to the FEC and application-level network coding. We then use Equations (1) and (3) to calculate the probability of the successful decoding

Algorithm 1 Reference Table Creation

```
Initialization: $h, j, k = 0$
Main:
for $k_b = 0$ to $k_b = (X - mn)/m$, step=$g$ do
    $h = h + 1$ // Index for reference table
    $k_p = X/(n + k_b) - m$
    for $\epsilon_b = 0.05$ to 1, $step = 0.05$ do
        $j = j + 1$ // Index for bit error probability
        for $\epsilon_I = 0.05$ to 1, $step = 0.05$ do
            $k = k + 1$ // Index for interference probability
            Use Equation (1) to calculate $\epsilon_e$
            $p = \epsilon_e + (1 - \epsilon_e) \times \epsilon_I$
            Use Equation (3) to calculate $\epsilon_s$
            $Ref(h, j, k) = \epsilon_s$
```

Algorithm 2 Search for Optimal Coding Scheme

```
Initialization: $max = 0$
Main:
for Each strategy $h$ in the reference table do
    $U = 0$
    for Each user $u_i$ do
        Search the reference table to find $\epsilon_{i,s}$ corresponding
        to $\epsilon_{i,b}$ and $\epsilon_{i,I}$
        $U = U + \epsilon_{i,s}$
        if $U > max$ then
            $max = U$
            Mark strategy $h$ as the selected policy
```

of the original m packets for each possible distribution and error rate. In order to reduce the size of the reference table and time complexity of the reference table creation, we limit the number of error rates that are considered and included in the reference table. For this purpose, we consider 0.05 as the granularity of the random bit error rate and interference probability. It should be noted that not only this granularity reduces the size of the reference table, but also decrease the time complexity of the second phase (search algorithm).

The details of the reference table creation algorithm are shown in Algorithm 1. In the first For loop, we change the FEC from 0 to $(X - mn)/m$ per packet. We increase the amount of FEC for each packet in step sizes equal to g. The remaining redundancy is used for the application-layer network coding. The number of linearly network coded packets that can be transferred is equal to $k_p = X/(n + k_b) - m$. In the second and third For loops, we change the bit error and interference probability, and use Equations (1), (2), and (3) to calculate the success probability. The algorithm returns a reference table for a given X, m, and n. The result will be a 3-dimensional reference table. The first dimension is the amount of FEC and application-layer network coding (coding strategy). Also, the second and third dimensions are the random bit error rate and the interference probability.

2) Search for Optimal Coding Scheme: As we mentioned in the previous subsection, the reference table needs to be generated only once, and the source node can use the reference table to find the optimal distribution in any delivery rate scenario. For each possible distribution of the transmissions, the source node searches the reference table to find the decoding probability of the coded packets that is correspondent to the destination's error rate. Assuming that the objective is defined as maximizing the summation of the successful decoding probabilities of all the destination nodes, the source node finds the distribution that maximizes this value. An advantage of our two-phases algorithm is that, we can easily modify the second phase of our proposed method (search phase) to consider other objective functions. For example, we can change the objective function to consider fairness or to guarantee a minimum successful decoding probability for each destination node. The details of the search algorithm are shown in Algorithm 2.

3) Unknown Channel: In the previous sections, we assumed that the channel condition is perfectly known by the source node. However, the channel condition of the wireless links is not fix, and changes over time. The total gain of the users is highly dependent on the bit error rate and the interference probability. As a result, the source node needs to learn them, when it does not have a perfect knowledge. In order to learn the channel conditions, each destination node needs to send periodic feedback messages to the source node, which contains the number of correctly received coded packets and the average number of bit errors. We represent the estimated bit error rate and the interference probability of user d_i at time τ as $\hat{e}_{i,b,\tau}$ and $\hat{e}_{i,I,\tau}$, respectively. Moreover, the number of bit errors and packets loss due to interference are represented as $r_{i,b,\tau}$ and $r_{i,I,\tau}$, respectively. The estimated bit error rate and the interference probability of user d_i at time $\tau + 1$ can be estimated as follows:

$$\hat{e}_{i,b,\tau+1} = \frac{(\tau - 1) \times \hat{e}_{i,b,\tau} + e_{i,b,\tau}}{\tau} \quad (4)$$

$$\hat{e}_{i,I,\tau+1} = \frac{(\tau - 1) \times \hat{e}_{i,I,\tau} + e_{i,I,\tau}}{\tau} \quad (5)$$

Here, $e_{i,b,\tau}$ and $e_{i,I,\tau}$ represent the bit error rate and the interference rate that user d_i experiences at time window τ, and can be calculated as follows:

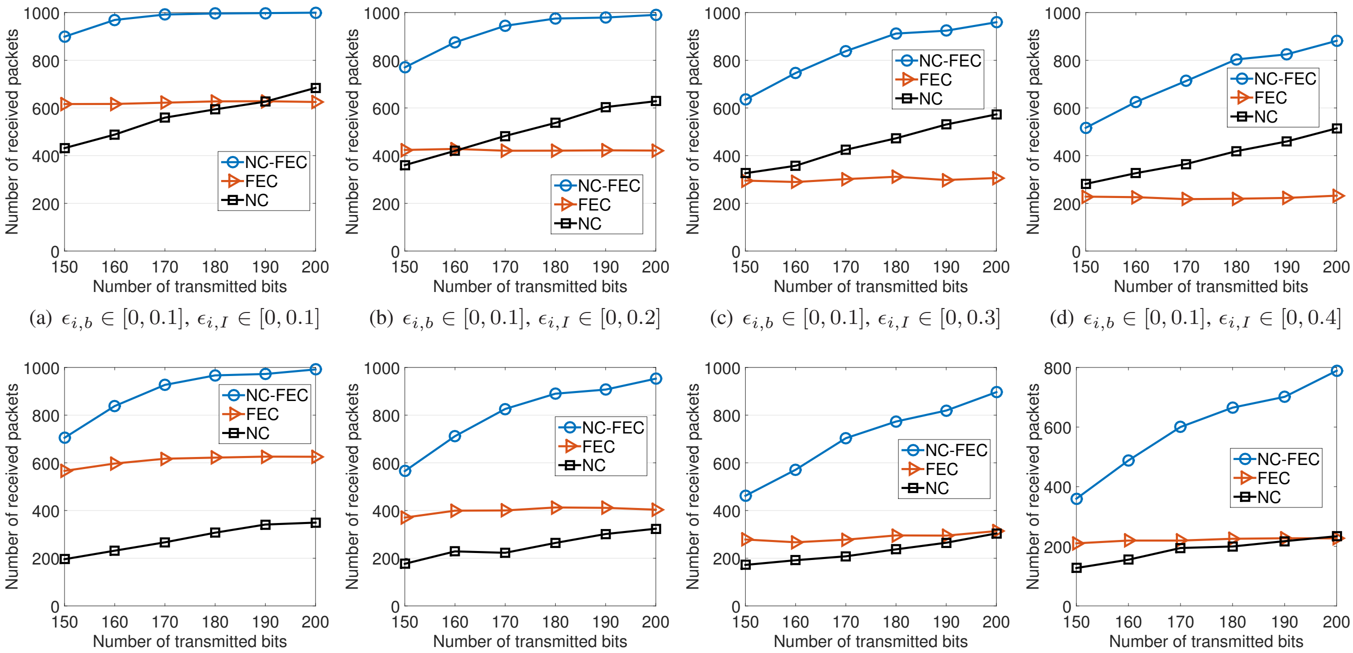
$$e_{i,b,\tau} = \frac{r_{i,b,\tau}}{(n + k_b)(m + k_p)}$$

$$e_{i,p,\tau} = \frac{r_{i,p,\tau}}{(m + k_p)}$$

In Equations (4) and (5), we multiply $\tau - 1$ by $\hat{e}_{i,b,\tau}$ and $\hat{e}_{i,I,\tau}$, respectively, to compute the total bit errors and the number of interferences in the previous $\tau - 1$ set of transmissions.

V. Evaluation

In this section, we evaluate our proposed method through extensive simulations. We first introduce the setting that is used in the evaluations. We then show and discuss the simulation results.



(a) $\epsilon_{i,b} \in [0, 0.1]$, $\epsilon_{i,I} \in [0, 0.1]$ (b) $\epsilon_{i,b} \in [0, 0.1]$, $\epsilon_{i,I} \in [0, 0.2]$ (c) $\epsilon_{i,b} \in [0, 0.1]$, $\epsilon_{i,I} \in [0, 0.3]$ (d) $\epsilon_{i,b} \in [0, 0.1]$, $\epsilon_{i,I} \in [0, 0.4]$

(e) $\epsilon_{i,b} \in [0, 0.2]$, $\epsilon_{i,I} \in [0, 0.1]$ (f) $\epsilon_{i,b} \in [0, 0.2]$, $\epsilon_{i,I} \in [0, 0.2]$ (g) $\epsilon_{i,b} \in [0, 0.2]$, $\epsilon_{i,I} \in [0, 0.3]$ (h) $\epsilon_{i,b} \in [0, 0.2]$, $\epsilon_{i,I} \in [0, 0.4]$

Fig. 4. Comparing the average number of received packets by the users in the case of transmitting 100 sets of packets; Number of users: 10; Original packet size: $n = 10$ bits; Number of packets to send in each set: $m = 10$.

A. Setting

In order to evaluate our proposed joint FEC and erasure code transmission scheme, we implemented a simulator in the MATLAB environment. We evaluate the performance of our algorithm by comparing it against transmitting the packets using only FEC and only random linear network coding methods, which we refer to them as FEC and the NC methods, respectively. In the FEC method, all the redundancy is assigned to the FEC. Also, in the NC method, the original packet are coded using random linear network coding, and the redundancy is assigned as the extra packet transmissions. We evaluate all of the methods on 1,000 topologies with random bit error ate and interference probability. The plots in this section are based on the average output of the simulations. We assume that the bit error rate and interference probability of the user nodes are independent, and are known by the source node. We evaluate the effect of bit error rate, interference probability, and the amount of redundancy on the number of received packets by the destination nodes. In the plots, we refer to our scheme as the NC-FEC method.

B. Results

In the first experiment, we study the effect of amount of redundancy on the number of received packets by the users. The number of users and the size of original packets are equal to 10. Moreover, the bit error rate and the interference probability of each user are set to a random value in the range of $[0, 0.1]$. The source node transmit 100 sets of packets. Each set of packets itself contains 10 packets. The number of assigned bit transmissions, which are shown in the plots are for each set of packets.

The average number of successfully received packets by the 10 users are shown in Figure 4(a). As expected, the number of successfully received packets in our proposed scheme is more than those in the FEC and the NC methods. It is clear that as the amount of redundancy increases, the number of successfully received packets increases as well. However, Figure 4(a) shows that the number of received packets in the FEC method is almost fixed. The reason is that, the bit error rate is not high in this evaluation. As a result, the 5 extra bits that are added to each original packet are enough for correcting the bit errors and assigning more FEC does not have any advantage. In contrast, the NC-FEC method assign the extra redundancy to the erasure code (NC) to combat the interference. In the case of 150 bit transmitting, the performance of the NC method is less than that of the FEC method. The reason is that, in the NC method, even a single bit error results in packet erasure. However, in the case of 200 bit transmissions, the extra transmitted packets can compensate this high erasure rate.

In Figure 4(b), we increase the interference probability to the range of $[0, 0.2]$ and repeat the previous experiment. The other settings are similar to that of Figure 4(a). The higher interference probability reduces the number of successfully received packets of all of the three methods. However, by adding more redundancy, the number of successfully received packets in our NC-FEC method approaches to 1000 very fast. Since the FEC method does not benefit from any inter-packet redundancy, it is more vulnerable to the interference. The plot shows that the number of successfully received packets in the

NC and FEC methods are up to 53% and 57% less that that of the NC-FEC method.

In Figure 4(c) and (d), we increase the interference probability to the ranges of $[0, 0.3]$ and $[0, 0.4]$, respectively, and repeat the previous experiment. The higher interference probability makes the number of successfully received packets of the FEC method less than that of in NC method. Comparing Figures 4(a)-(d), we can find that our proposed NC-FEC method is more robust against the interference probability that the FEC and the NC methods. The reason is that, the NC-FEC method uses the redundancy more efficiently that the other methods.

In the next set of four experiments, we increase the range of bit error rate to $[0, 0.2]$, and repeat the previous experiments. The number of received packets of the FEC method in Figure 4(e) is up to 3 times that of the NC method. The reason is that, the NC method is vulnerable to even a single bit error. The high bit error rate in Figure 4(e) results in too many packet erasures, which cannot be compensated by the added inter-packet redundancy of the NC method. The number of received packets of the NC and FEC methods in Figure 4(a) are up to 72% and 37% less than that of the NC-FEC method.

From Figures 4(a)-(h), we find that as the bit error rate and interference probability increases, both of the NC and FEC methods become feeble. In Figure 4(h), the number of received packets in both of the NC and FEC methods are very close, and both of the methods perform very poor. In this case, increasing the number of transmissions even to 200 bits does not have a great impact on the number of received packets of the FEC and the NC methods. In contrast, our proposed method performs well even in the case of high bit error rate and interference probability. In Figure 4(h), the number of successfully received packets in our scheme is up to 4 times that of the FEC and the NC method

In the next set of experiments, we study the effect of interference on the number of received packets. There is a single user in the network, and there are 100 sets of 10 packets to be transmitted. The size of each packet before adding FEC is equal to 10 bits. In Figure 5(a), the bit error rate of the user is randomly set to a value in the range of $[0, 0.1]$. Also, the number of transmitted bits is set to 200. The interference range is shown in the caption of the X-axis. As Figure 5(a) shows, the NC-FEC and the FEC methods have the highest and the lowest number of received packets, respectively. Also, the FEC method is highly vulnerable to the interference, which is due to lake on any inter-packet erasure code. For an interference probability in the range of $[0.2, 0.3]$, only about 60 out of the 1000 packets are received correctly. The figure shows that our proposed NC-FEC method is more robust to the interference compared to the other methods.

In Figure 5(b), we increase the number of transmitted bits to 220, which results in more received packets in the case of using NC or NC-FEC methods. However, this increase in the redundancy does not have much effect on the FEC method. The figure shows that in the case of interference in the range of $[0.2, 0.3]$, the number of received packets in the NC-FEC method is about 12 times that of the NC method.

We repeat the experiment in the case of 240 and 260 bit transmissions in Figures 5(c) and (d), respectively. Figures 5(a)-(d) show that the number of received packets in the case of using the FEC method and with an interference probability on the rang of $[0.4 - 0.5]$ is almost zero. Moreover, the number of transmitted bits does not have much impact on the number of received packets in the case of the FEC method.

In the next set of four experiments, we increase the bit error rate to the range of $[0.1, 0.2]$, and repeat the last four experiments. The results are shown in Figures 5(e)-(h). Due to lake of intra-packet redundancy, the NC method becomes too fragile in the case of high bit error rate. Figure 5(h) shows that even with 260 bit transmissions, at most only 5 packets are delivered correctly by the NC method. Also, for an interference probability in the range of [0.4,0.5], the NC and FEC methods cannot deliver any packet successfully. In this case, our proposed NC-FEC method can still deliver about 530 packets successfully.

VI. Conclusion

Due to unreliability of the wireless links, providing reliable transmissions is a challenge in wireless networks. Noises and interference among the wireless nodes are two main sources of errors in wireless transmissions. In order to address the errors due to noise, forward error correction methods are widely used. However, in the case that the network is too noisy, or there is interference among the wireless nodes, application-layer erasure codes, such as network coding, are useful. In this paper, we address the problem of providing robust data transmission in a one-hop multicast network.

In order to provide a robust and reliable wireless transmissions, we benefit from joint forward error correction and erasure coding in our proposed method. We derive the relation of bit error rate and interference with successful delivery of the packets. In order to find the optimal distribution of the available redundancy among FEC and erasure code, we propose a low-complexity two-phases method, which has offline and online phases. In the first phase, a reference table is generated, which shows the successful delivery of the packets for each possible redundancy distribution. In the second phase, the source node uses this reference table to find the optimal strategy depending on the noise and interference level. In order to evaluate our proposed method and show its effectiveness, we preform extensive simulation.

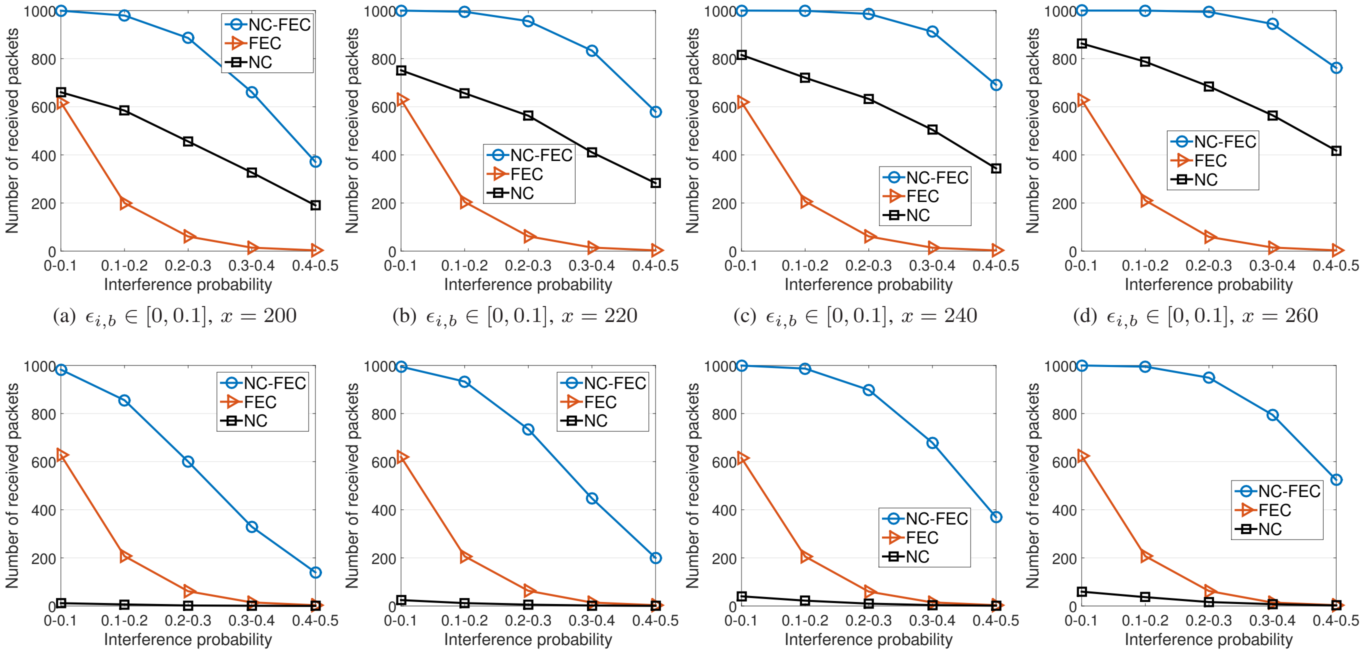
Acknowledgment

This work is supported in part by NSF grants CNS 149860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963.

References

[1] H. Djandji, “An efficient hybrid arq protocol for point-to-multipoint communication and its throughput performance,” *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1688–1698, 1999.

[2] G. C.Clark and J. B. Cain, *Error-correction coding for digital communications*. Springer, 1981.



(a) $\epsilon_{i,b} \in [0, 0.1]$, $x = 200$ (b) $\epsilon_{i,b} \in [0, 0.1]$, $x = 220$ (c) $\epsilon_{i,b} \in [0, 0.1]$, $x = 240$ (d) $\epsilon_{i,b} \in [0, 0.1]$, $x = 260$

(e) $\epsilon_{i,b} \in [0.1, 0.2]$, $x = 200$ (f) $\epsilon_{i,b} \in [0.1, 0.2]$, $x = 220$ (g) $\epsilon_{i,b} \in [0.1, 0.2]$, $x = 240$ (h) $\epsilon_{i,b} \in [0.1, 0.2]$, $x = 260$

Fig. 5. Evaluating the effect interference of the number of received packets in the case of transmitting 100 sets of packets to a single user; Original packet size: $n = 10$ bits; Number of packets to send in each set: $m = 10$.

[3] S. Lin and D. J. Costello, *Error control coding: Fundamentals and Applications*. Prentice-hall Englewood Cliffs, NJ, 2004.

[4] W. Ryan and S. Lin, *Channel codes: classical and modern*. Cambridge University Press, 2009.

[5] B. Zhao and M. Valenti, "Practical relay networks: a generalization of hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, 2005.

[6] ——, "The throughput of hybrid-ARQ protocols for the gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1971–1988, 2001.

[7] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[9] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[10] P. Ostovari, J. Wu, and A. Khreishah, *Network Coding Techniques for Wireless and Sensor Networks*. Springer, 2013.

[11] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

[12] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[13] P. Cataldi, M. Shatarski, M. Grangetto, and E. Magli, "Lt codes," in *IIH-MSP'06*, 2006, pp. 263–266.

[14] G. Angelopoulos, A. Paidimarri, A. P. Chandrakasan, and M. Médard, "Experimental study of the interplay of channel and network coding in low power sensor applications," in *IEEE ICC*, 2013, pp. 5126–5130.

[15] L. Rizzo and L. Vicisano, "RMDP: an FEC-based reliable multicast protocol for wireless environments," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 2, pp. 23–31, 1998.

[16] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.

[17] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ecc energy-efficient?" *EURASIP Journal on Wireless Communications and Networking*, vol. 17, no. 2, pp. 29–29, 2006.

[18] M. Vuran and I. F. Akyildiz, "Error control in wireless sensor networks: a cross layer analysis," *ACM Transactions on Networking, IEEE*, vol. 17, no. 4, pp. 1186–1199, 2009.

[19] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM*, 2006, pp. 243–254.

[20] D. Koutsonikolas, C. Wang, and Y. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *IEEE INFOCOM*, 2010, pp. 1–9.

[21] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SIGCOMM*, 2007.

[22] P. Ostovari, J. Wu, and A. Khreishah, "Deadline-aware broadcasting in wireless networks with local network coding," in *IEEE ICNC*, Jan 2012.

[23] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[24] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[25] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782– 795, Oct 2003.

[26] L. Lu, M. Xiao, M. Skoglund, L. Rasmussen, G. Wu, and S. Li, "Efficient network coding for wireless broadcasting," in *Wireless Communications and Networking Conference (WCNC)*, 2010, pp. 1–6.

[27] L. Lu, M. Xiao, and L. Rasmussen, "Relay-aided broadcasting with instantaneously decodable binary network codes," in *ICCCN*, 2011, pp. 1–5.

[28] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.

[29] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 401–412, 2008.

[30] M. Li, Z. Yang, and W. Lou, "Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 223–235, 2011.

[31] Z. Yang, M. Li, and W. Lou, "CodePlay: Live multimedia streaming in vanets using symbol-level network coding," in *ICNP*, 2010, pp. 223–235.