Symbol-Level Reliable Broadcasting of Sensitive Data in Error-Prone Wireless Networks

Pouya Ostovari[a,*], Jie Wu[a], Abdallah Khreishah[b]

[a] *Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122*

[b] *Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102*

Abstract

Reliable packet transmission over error-prone wireless networks has received a lot of attention from the research community. In this paper, instead of using simple packet retransmissions to provide reliability, we consider a novel retransmission approach, which is based on the importance of bits (symbols). We study the problem of maximizing the total gain in the case of partial data delivery in error-prone wireless networks, in which each set of bits (called symbols) has a different weight. We first address the case of one-hop single packet transmission, and prove that the optimal solution that maximizes the total gain has a round-robin symbol transmission pattern. Then, we extend our solution to the case of multiple packets. We also enhance the expected gain using random linear network coding. Our simulation results show that our proposed multiple packets transmission mechanism can increase the gain up to 60%, compared to that of a simple retransmission. Moreover, our network coding scheme enhances the expected total gain up to 15%, compared to our non-coding mechanism.

Keywords: Symbol-level coding, broadcasting, reliability, random linear network coding, weight, wireless networks, error-prone channel.

1. Introduction

Broadcasting schemes are widely used for disseminating data and control messages in wireless networks. However, the error-prone wireless links creates challenges in these networks. To handle these challenges, different mechanisms [1–5] have been proposed to provide reliability. In the case of numeric data, e.g., the captured information by sensor nodes, the importance of the data (numbers) decreases from the left (most significant bit) to the right (least significant bit). Therefore, any mechanism that addresses numeric data transmissions

*Corresponding author

Email addresses: ostovari@temple.edu (Pouya Ostovari), jiewu@temple.edu (Jie Wu), abdallah.khreishah@njit.edu (Abdallah Khreishah)

in a lossy environment should consider the weights of the bits. The problem of reliable transmission has received a lot of attention; however, to the best of our knowledge, nobody has studied the problem of transmitting symbols (a group of bits) with different weights.

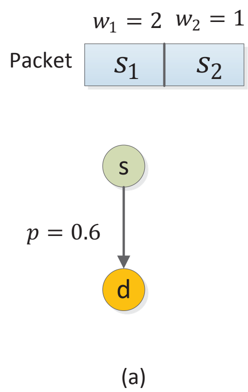
In contrast to the previous works, in this paper, we propose a novel broadcasting approach in wireless networks which considers the importance of the symbols. Instead of providing reliable transmissions and guaranteeing a full delivery of the data, we are interested in maximizing the expected total gain of the destination nodes, with a fixed given number of symbol transmissions. In applications such as transmitting numeric data from a source node to a set of destination nodes, encountering an error in more important bits has a more negative impact, and with a given number of transmissions, it is more efficient to allocate more transmissions to the most important part of the data.

Figure 1 (a) shows an example, in which a packet with 2 symbols is transmitted to a destination node. The weights of the symbols s_1 and s_2 are equal to 2 and 1, respectively. Assume that the error-rate of the link is equal to 0.6. The window size for transmitting the packet is equal to 2 symbols, and after that, another packet will be ready for transmission. In this case, the traditional methods transmit each symbol once. Now, let us compute the expected gain. We represent the number of transmissions of symbols s_1 and s_2 as x_1 and x_2, respectively. Thus, the probability of successful delivery of symbols s_1 and s_2 is equal to $1 - p^{x_1}$ and $1 - p^{x_2}$, respectively. Consequently, the expected gain is equal to $w_1 \times (1 - p^{x_1}) + w_2 \times (1 - p^{x_2})$, where w_1 and w_2 are the weights of symbols s_1 and s_2. The possible distribution of 2 transmissions and their respective utilities are shown in Figure 1 (b). The figure shows that it is more efficient to allocate both of the transmissions to symbol s_1. Now assume that the window size is equal to 3 transmissions. Figure 1 (b) shows that the optimal solution is allocating 2 transmissions to symbol s_1, and 1 transmission to symbol s_2. It should be noted that if there is no deadline, then the optimal solution is a simple extension from the channel coding theory [6].

Finding the importance of a data is application specific. As another example, consider a multi-layer (multi-resolution) video [7–9]. In multi-layer video coding, each video is divided into a base layer and a set of enhancement layers. The base (first) layer is required to watch the video. In contrast, the enhancement layers can increase the quality of the video. However, a layer is not useful without the layers with a smaller index. In this case, the layers with a smaller index are more important than the layers with a greater index. In order to assign weights to the different layers, we can measure the effect (quality enhancement) of adding a layer to the layers with a smaller index and consider it as the weight of that layer.

In this work, we answer the following question. How should we distribute the transmissions to different symbols with unequal importance in order to maximize the total expected gain? While answering this question, we have the following contributions:

- In contrast to previous works, which study the problem of reliable packets



2 transmissions

x_1	x_2	Utility
2	0	1.28
1	1	1.2
0	2	0.64

3 transmissions

x_1	x_2	Utility
3	0	1.568
2	1	1.68
1	2	1.44
0	3	0.78

(b)

Figure 1: Motivation example; (a) setting, (b) the choices with 2 and 3 transmissions.

or symbol level transmission, we study the problem of maximizing the total gain in the case of partial data delivery.

- In the case of single packet transmission to multiple destinations with homogeneous channel conditions, we propose an algorithm to find the optimal solution, and prove its optimality. This algorithm assigns the transmissions to the symbols in a set of round-robin iterations.

- We also propose an optimal algorithm for the case of transmitting a single packet to multiple destinations with heterogeneous channels.

- We extend the proposed single packet transmission algorithms to the case of multiple packets, and use the advantage of random linear network coding to enhance the expected gain.

- We show that network coding does not necessarily increase the gain, and we find the condition that network coding results in more gain than the non-coding mechanism.

The rest of this paper is organized as follows. Section 2 reviews the related work and describes linear network coding. In Section 3, we provide the problem definition and the setting. We propose our mechanisms for the case of transmitting a single packet in Section 4. In Section 5, we extend our proposed mechanism to the case of transmitting multiple packets, and we boost the gain of the proposed method using linear inter-packets network coding. We discuss the implementation issues in Section 6, and evaluate the proposed mechanisms through simulations in Section 7. Section 8 concludes the paper.

2. Related Work and Background

2.1. Reliable Transmission

Certain mechanisms, such as feedback messages, can be applied in error-prone wireless networks to provide reliability. Automatic Repeat reQuest (ARQ) is one of the most frequently used approaches for addressing this challenge [1]. Nevertheless, ARQ imposes overhead, since it requires transmitting many feedback messages, especially for the case of multi destination nodes. Hybrid-ARQ approaches [2, 10], which combine FEC (Forward Error Correction) with ARQ, are proposed to solve this problem. The RMDP approach, which is a complex method, [10] uses Vandermonde [11] code and ARQ to ensure reliability.

Using rateless (fountain) codes [3–5] is an efficient way to provide reliability without using feedback messages. In these schemes, the source node can generate and transmit an unlimited number of encoded packets until each destination node receives enough encoded packets to retrieve the original packets. In this scheme, the destination nodes need to collect a sufficient number of encoded packets, regardless of which packets have been lost. Assuming that the number of original packets is k, the number of sufficient coded packets that need to be received is $N = (1 + \epsilon)$ [3], where ϵ is a small number and shows the overhead of the rateless codes. Note that ϵ is independent of the reliability of the links. It can be shown that as $k \to \infty$, the overhead goes to zero [12]. Therefore, rateless codes are very efficient for transmitting a large number of packets, but are inefficient for transmitting a small number of packets. As a result, rateless codes are not appropriate for delay-sensitive applications, such as our problem, which needs small batches of packets.

2.2. Network Coding

Network coding (NC) [13–19] is introduced in [20] for wired networks, to solve the bottleneck problem in single multicast problem. It is shown in [21] that linear network coding achieves the capacity for the single multicast session problem. The authors in [22] provide a useful algebraic representation of the linear network coding problem. *Random linear network coding* is proposed in [23], and it is shown that randomly selecting the coefficients of the coded packets, achieves the capacity asymptotically, with respect to the finite field size.

In random linear network coding, coded packets are the random linear combination of the original packets over a finite field. The coded packets are in the form of $\sum_{i=1}^{k} \alpha_i \times P_i$, where P and α are the packets and random coefficients, respectively. Using random linear network coding, the source node generates and transmits random coded packets and their respective random coefficient vector. The destination nodes are able to decode the coded packets once they receive k linearly independent coded packets. The decoding process is done using Gaussian elimination for solving a system of linear equations. Using this scheme, the destination nodes can send just one acknowledgment message to stop the source node from sending more coded packets once they are able to decode the coded packets.

The work in [24–27] address the problem of reliable one-hop broadcasting. In order to provide reliability, the source node needs to retransmit the lost packets by the destination nodes. The source node uses the benefit of network coding in the retransmission phase to improve the transmission efficiency. In order to reduce the number of required retransmissions, these methods combine the packets that have not been received correctly by different receiver nodes. Assume that in Figure 2, the source node sends packets P_1 and P_2, and destination nodes d_1 and d_2 only receive packets P_1 and P_2, respectively. As a result, the source node needs to retransmit both of the packets. However, using network coding, the source node can mix the packets to send a single packet $P_1 + P_2$. If nodes d_1 and d_2 receive the coded packets, they can retrieve their respective packets P_2 and P_1, by performing $(P_1+P_2)-P_1$ and $(P_1+P_2)-P_2$, respectively.

Symbol-level network coding for wireless mesh networks is introduced in [28], and it is shown that its throughput is more than that of the packet-level network coding. The insight behind the symbol-level coding is that, even in the case that a node does not receive a coded packet correctly, some of the symbols that form the packet might be received without any error. As a result, if instead of coding the packets together we code the symbols, the successfully received symbols do not need to be retransmitted, and transmitting the remaining symbols is sufficient; therefore, symbol-level transmission reduces the transmission cost.

The authors in [29, 30] use the symbol-level coding to propose a method for distributing data and multimedia in vehicular networks. They show that the symbol-level network coding outperforms the packet level network coding for content distribution in Vehicular Ad-Hoc Networks (VANET). The goal in [30] is to efficiently designate live streaming multimedia to the mobile nodes in a specific region of a road, called an area of interest.

3. Setting

We consider a single-hop wireless network that consists of one source and n destination nodes d, as depicted in Figure 2. The source node has a batch of k packets to send to the destination nodes, and each packet consists of m symbols. Each symbol itself might contain several bits. Each symbol has a weight w_i, and in general, $w_i > w_{i+1}, \forall i : 1 \leq i \leq m-1$. For simplicity, we assume that the weight of the i-th symbols of all of the packets are the same. However, the proposed solutions in this paper can be easily extended to the case of packets with different symbols' weights. We assume that the error rate of each transmitted symbol (or packet) from the source node to the i-th destination node is equal to p_i. We represent the number of times that the i-th symbol is transmitted as x_i.

In our model, the packets of a batch have a deadline to be received by the destination nodes, which is equal to the window size, and after this time another batch of packets will be ready for transmission. As a result, channel coding, hierarchical coding, and unequal error protection methods cannot be applied in our setting. We assume that this window size for a batch of packets is enough for transmitting $t \times k$ symbols, where t is the assigned window for a

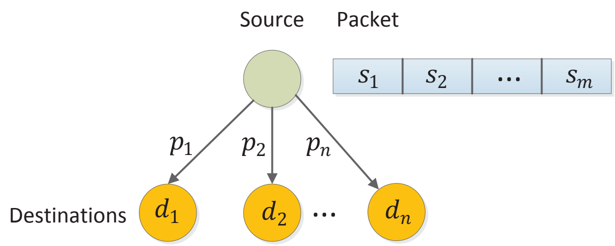


Figure 2: System setting.

single packet. If the packets are not delay sensitive, or the source has infinite packets to transmit, the optimal solution is a simple extension of the well-known channel coding theory [6]. Our goal is to maximize the total weight of the received symbols of a batch of k packets by the destination nodes. As a result, our utility function becomes:

$$u = k \times \sum_{i=1}^{m} \sum_{l=1}^{n} w_i \times (1 - {p_l}^{x_i}) \tag{1}$$

$$s.t. \quad \sum_{i=1}^{m} x_i = t$$

It is obvious that we should assign a larger portion of the transmissions to the symbols that are more important than the other symbols, as successful delivery of these packets to the destination nodes results in more gain. However, it is not clear how we should assign and distribute the transmissions (duplications) to the different symbols of the packets in order to maximize the total gain. Our goal in this work is to find this optimal assignment. In the rest of the paper we use gain and utility interchangeably. The set of symbols used in this paper is summarized in Table 1.

For the case of data like binary data, in which the importance of the i-th bit is twice that of the $i+1$-th symbol, the weight of the i-th symbol can be defined as 2^{m-i}. As a result, the objective function becomes:

$$k \times \sum_{i=1}^{m} \sum_{l=1}^{n} 2^{m-i} \times (1 - {p_l}^{x_i})$$

In this case each symbol contains one bit. In Binary-Coded Decimal (BCD), each decimal digit is represented with a fixed number of bits, usually 4 bits. Figure 3 shows a decimal number and its conversion to BCD. For the case of BCD, we can consider the 4 bits that correspond to the same decimal digit as a symbol, in which the weight of the i-th symbol is 10 times that of the $i+1$-th

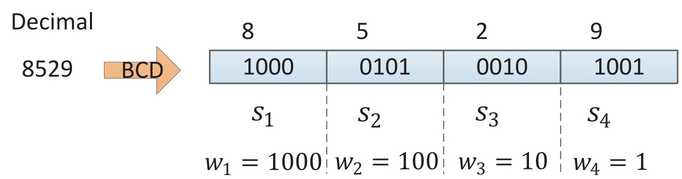


Figure 3: Binary coded decimal (BCD) and the weights of the symbols.

symbol. Consequently, for the BCD data, the objective function will be:

$$k \times \sum_{i=1}^{m} \sum_{l=1}^{n} 10^{m-i} \times (1 - {p_l}^{x_i})$$

In this work, we do not restrict the solution to a special weighting system, and solve the problem in the general case. The parameters in our proposed method can be adjusted based on the application and the structure of the data to be transmitted.

4. Optimal Solution for the Case of Single Packet

In the following two sections, we first find the optimal distribution of transmissions to different symbols in the case of destination node with homogeneous channels, and then we extend it for heterogeneous destination nodes.

4.1. destinations with Homogeneous Channels

We first investigate and address the problem in the case of a packet size equal to 2 symbols. Then, we generalize the solution to the case of m symbols.

4.1.1. Packet Size $m = 2$

for a packet size $m = 2$, the objective function becomes:

$$\begin{aligned} &u = n \times \left[w_1 \times (1 - p^{x_1}) + w_2(1 - p^{x_2}) \right] \\ s.t. \quad &x_1 + x_2 = t \end{aligned}$$

We denote the change in the total gain as we increase the i-th symbol's transmissions from x_i to $x_i + 1$ as Δ_{x_i}, so we have:

$$\begin{aligned} \Delta_{x_i} &= n \times w_i \times (1 - p^{x_i+1} - (1 - p^{x_i})) \\ &= n \times w_i \times (1 - p) \times p^{x_i} \end{aligned}$$

Table 1: The set of symbols used in this paper.

Notation	**Definition**
d_i	The i-th destination node
n	The number of destination nodes
m	The number of symbols inside each packet
k	The number of packets
w_i	The weight of the i-th symbol of each packet
p_i	The error rate of the link between the source and the i-th destination node.
t	The size of the transmission time window for each packet (in the term of number of symbols)
s_i	The i-th symbol (in the case of single packet)
$s_{j,i}$	The i-th symbol of the j-th packet
S_i	The i-th coded symbol
P_i	The i-th packet
Δ_{x_i}	The change in the utility gain as we increase x_i to $x_i + 1$
u	The utility function
u_i	The gain (utility) from the i-th symbols
u_i^{NC}	The gain (utility) from the i-th symbols when we use linear NC
u^{NC}	The total gain (utility) of using linear NC
u^{UC}	The total gain without using linear NC
c^{NC}	The header cost of a linear coded packet
c^{UC}	The header cost of an uncoded packet
τ	The number of sets of t transmissions performed so far
$p_{i,\tau}$	The actual error rates of the link between the source and node d_i in the set of τ-th set of transmissions
$\hat{p}_{i,\tau}$	The estimated error rates of the link between the source and node d_i in the set of τ-th set of transmissions
$r_{i,\tau}$	number of successfully received symbols by the destination node d_i in the set of τ-th set of transmissions

As mentioned in the setting, $w_1 > w_2$. Thus, it is obvious that, in order to achieve more gain, the number of times the source node transmits the first symbol should be more than or equal to that of the second symbol. If we consider the problem in t rounds of transmission, the first time we should increment x_2 and transmit the second symbol is when the gain of increasing x_1 is less than that of increasing x_2. In other words, the condition to increase x_2 is $\Delta_{x_1} < \Delta_{x_2}$. Consequently, we have:

$$n \times w_1 \times (1-p)p^{x_1} < n \times w_2 \times (1-p)p^{x_2}$$

and as a result,

$$p^{x_1} < \frac{w_2}{w_1} p^{x_2} \tag{2}$$

In this case, we are incrementing x_2 for the first time, so $x_2 = 0$, and we have:

$$p^{x_1} < \frac{w_2}{w_1} \tag{3}$$

Therefore, the first time we should increment x_2 is when $p^{x_1} < \frac{w_2}{w_1}$; we refer to this point as the *saturation point*. After this point, whenever $p^{x_1} < \frac{w_2}{w_1} p^{x_2}$, we should increment x_2, since it results in more gain. In contrast, if $p^{x_1} \geq \frac{w_2}{w_1} p^{x_2}$, we increment x_1.

We show the optimal distribution of the transmissions between x_1 and x_2 for different total numbers of transmissions t in Figure 4. The weights of symbols s_1 and s_2 in this example are assumed to be 5 and 1, respectively. To find the optimal distribution, we compute the utility for all of the possible distributions. It can be inferred from this figure that, after incrementing x_2 for the first time, the optimal solution has a round-robin incrementing pattern. The insight behind this phenomenon is as follows. The ratio of Δ_{x_1} and Δ_{x_2} is equal to:

$$\frac{n \times \Delta_{x_1}}{n \times \Delta_{x_2}} = \frac{w_1 \times (1-p) \times p^{x_1}}{w_2 \times (1-p) \times p^{x_2}} = \frac{w_1 \times p^{x_1}}{w_2 \times p^{x_2}} \tag{4}$$

Before we reach the saturation point, $\Delta_{x_1} \geq \Delta_{x_2}$, and the ratio in Equation (4) is greater than 1. However, after the saturation point, whenever we increment x_2, the ratio in Equation (4) is multiplied by $\frac{1}{p}$, and it becomes greater than 1. As a result, the next transmission should be assigned to x_1, as it results in more gain. In contrast, whenever we increment x_1, the ratio is multiplied by p, which makes the ratio less than 1. In this case, it is more beneficial to assign the next transmission to x_2.

Based on the discussion, our algorithm works as follows. We iteratively increment x_1 and decrement t until $p^{x_1} < \frac{w_2}{w_1}$. If any more transmissions are left, we start to distribute these remaining transmissions between x_1 and x_2 in a round-robin pattern. We prove the optimality of this algorithm in the Appendix Appendix A.1.

4.1.2. General Packet Size m

Similar to the case of $m = 2$, the first symbol (the symbol with the smallest index) has more weight, so it is more important than the other symbols. As a result, we should not transmit other symbols until $\Delta_{x_1} > \Delta_{x_2}$. It should be noted that this condition implies that $\Delta_{x_1} > \Delta_{x_i}, \forall i : 2 \leq i \leq m$. The reason is that, $w_2 > w_i, \forall i : 3 \leq i \leq m$, and $x_i = 0, \forall i : 2 \leq i \leq m$. Consequently, similar to the case of packet size $m = 2$, the first time that we should increment x_2 is when $p^{x_1} < \frac{w_2}{w_1}$, and after this point, the transmissions should be distributed between x_1 and x_2. However, in contrast with the case of $m = 2$, after a specific

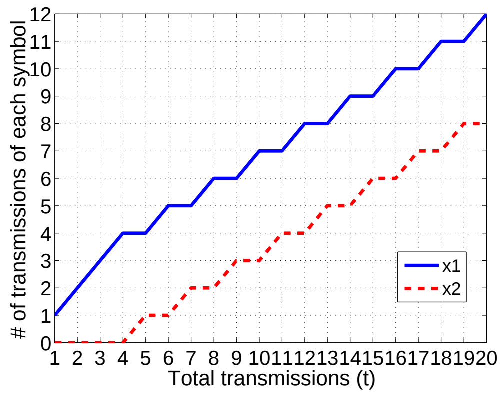


Figure 4: Optimal distribution of transmissions between 2 symbols for an error probability $p = 0.5$, $w_1 = 5$, and $w_2 = 1$.

point, we should start to transmit the third symbol. The condition to increment x_3 is when $\Delta_{x_1} < \Delta_{x_3}$ and $\Delta_{x_2} < \Delta_{x_3}$. For $\Delta_{x_1} < \Delta_{x_3}$ we have:

$$n \times w_1 \times (1-p)p^{x_1} < n \times w_3 \times (1-p)p^{x_3}$$

At this step, we are increasing x_3 for the first time; therefore, $x_3 = 0$, and the first optimality condition becomes $p^{x_1} < \frac{w_3}{w_1}$. Moreover, for the second condition $\Delta_{x_2} < \Delta_{x_3}$ we have:

$$n \times w_2 \times (1-p)p^{x_2} < n \times w_3 \times (1-p)p^{x_3}$$

As $x_3 = 0$, the equation becomes $p^{x_2} < \frac{w_3}{w_2}$. When these two conditions hold, we should start assigning the remaining transmissions to the first 3 symbols in a round-robin pattern. By the same reasoning, the condition for increasing x_m is when $p^{x_i} < \frac{w_m}{w_i}, \forall i : 1 \le i \le m-1$. Figure 5 shows the optimal distribution of the transmissions between different symbols when $m = 5$ for different numbers of total symbol transmissions t. The link's error rate and w_i are equal to 0.5 and 2^{5-i}, respectively. This figure shows that, even in the case of a packet size more than 2 symbols, the round-robin distribution of the transmissions results in the optimal solution.

We can summarize the discussion and the procedure of our weighted retransmission with homogenous destinations (WRH) algorithm as follows. We assign the transmissions to x_1 until $p^{x_1} < \frac{w_2}{w_1}$. Then, we distribute the remaining transmissions between x_1 and x_2 until $p^{x_1} < \frac{w_3}{w_1}$ and $p^{x_2} < \frac{w_3}{w_2}$. After this point, we continue the round-robin distribution of the remaining transmissions among x_1, x_2, and x_3. In general, we start incrementing x_j when $p^{x_i} < \frac{w_j}{w_i}, \forall i : 1 \le i \le j-1$, and we add x_j to the round-robin incrementing process. We continue this process until t becomes 0. The proof of this algorithm's optimality is presented in Appendix Appendix A.2.

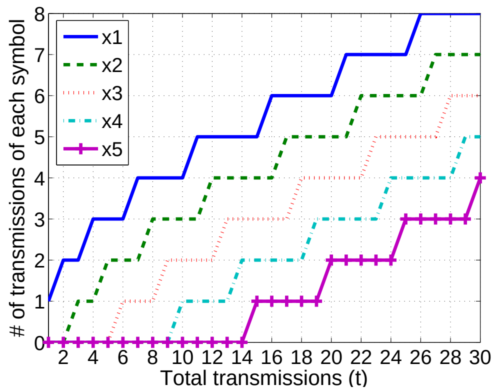


Figure 5: Optimal distribution of transmissions between 5 symbols for an error probability $p = 0.5$, $w_1 = 16$, $w_2 = 8$, $w_3 = 4$, $w_4 = 2$, and $w_5 = 1$.

The binary and BCD representations of decimal number 83 are shown in Figures 6 (a) and (b), respectively. In BCD, the weight of symbol s_1 is 10 times that of symbol s_2. Also, the weight of symbol s_i is twice that of symbol s_{i+1} in binary representation. Assuming that the error rate of the link between the source and destination nodes is equal to 0.2, we show the optimal solutions in the cases of 8, 12, and 16 symbol transmissions for the binary number in Figure 6 (a). Note that, in this case, each symbol contains one bit. In BCD, the size of each symbol is 4 times that of the binary representation. Therefore, in Figure 6 (b), we show the optimal transmissions with 2, 3, and 4 symbol transmission in the BCD representation.

4.2. Destinations with Heterogenous Channels

In the case of multiple destination nodes with different transmission error rates, the round-robin distribution pattern does not exist. For this reason, we use an iterative algorithm, which we call weight retransmission (WR). In each iteration of the WR method, we assign one transmission to a symbol such that it maximizes the increase in the total gain. In the case of heterogenous destination nodes, Δ_{x_i} can be calculated as follows:

$$\begin{aligned}\Delta_{x_i} &= w_i \times \sum_{l=1}^{n} \left[1 - p_l^{x_i+1} - (1 - p_l^{x_i})\right] \\ &= w_i \times \sum_{l=1}^{n} \left[p_l^{x_i} - p_l^{x_i+1}\right]\end{aligned}$$

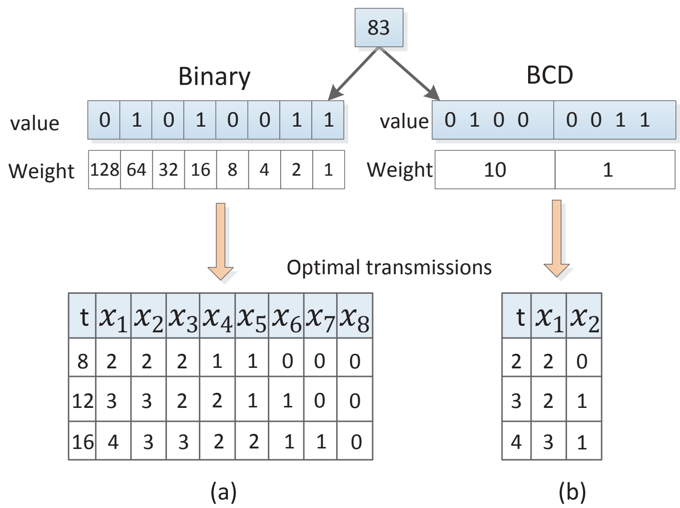


Figure 6: The binary and BCD representations of a decimal number, and the optimal symbol transmissions to homogeneous destinations with a different total number of symbol transmissions. $p = 0.2$; (a) Binary representation, (b) BCD representation

and the total utility is equal to:

$$u = \sum_{i=1}^{m} \left[w_i \times \sum_{l=1}^{n} (1 - p_l^{x_i}) \right]$$

The WR algorithm assigns the total number of transmissions t to the different symbols in t rounds. At each iteration, our algorithm computes $\Delta_{x_i}, \forall i : 1 \leq i \leq m$, and it assigns the current transmission to x_j that increasing its number of transmissions by one results is more increase in the total gain. In other words, $j = \arg\max_{1 \leq i \leq m} \Delta_{x_i}$. Algorithm 1 shows the iterative process.

The second loop (the loop over j) and its internal for loop in Algorithm 1 run t and m times, respectively. Moreover, Δ_{x_i} is a summation over n nodes. As a result, the complexity of the WR method is in the order of $O(t \times m \times n)$. We leave the proof of optimality to Appendix Appendix B.

5. Efficient Solution for the Case of Multiple Packets

In order to broadcast a batch of k packets from a source node to a set of destination nodes, we can use two approaches: without and with network coding. We describe the details of the mechanisms in the following sections.

Algorithm 1 WR Algorithm

```
for i=1 to m do
    x_i = 0
for j=1 to t do
    max = 0
    argmax = 0
    for i=1 to m do
        Δ_{x_i} = w_i × Σ_{l=1}^{n} (p_l^{x_i} − p_l^{x_i+1})
        if Δ_{x_i} > max then
            max = Δ_{x_i}
            argmax = i
    x_argmax = x_argmax + 1
```

5.1. Without Network Coding

In our model, we assume that the packet sizes (in term of symbols) are equal. Moreover, the weights of the i-th symbols in different packets are the same. As a result, the problem of sending k independent packets becomes k similar problems with the same solution. Consequently, we can simply use the result of the previous section, and repeat the same process for the different packets. In the weighted multiple packets retransmission (WMPR) mechanism, we first compute the optimal number of transmissions for each symbol. For this purpose, we perform one of the proposed algorithms in the previous section (WRH or WR), depending on the channels condition. Then, we use the output values x_i from the first step, and transmit each of the i-th symbols of the different packets x_i times. As we repeat the same process on k packets, the utility of this scheme is k times the gain of transmitting one packet.

5.2. Inter-Packet Network Coding

Random linear network coding can increase the gain of the WMPR mechanism. In our heuristic algorithm with network coding, much similar to the WMPR method, we run the WRH or WR algorithms to compute the optimal value of $x_i, \forall i : 1 \leq i \leq m$. Then, as it is shown in Figure 7, we code all of the i-th symbols of the k packets together. We denote the i-th coded symbols, as S_i. The coded symbols are in the form of $S_i = \sum_{j=1}^{k} \alpha_j \times s_{j,i}$, where $\alpha_{j,i}$ is a random coefficient. In this scheme, the source node generates and sends $x_i \times k$ coded symbols from the i-th original symbols. This is in contrast with the WMPR approach, in which the source node transmits the i-th symbol of each packet x_i times ($x_i \times k$ transmissions for k packets). We refer to our proposed weighted multiple packets retransmission method with network coding as WMPR-NC method.

In the discussed inter-packet network coding policy, each destination node is able to decode the i-th coded symbols and retrieve the k original i-th symbols of different packets, if it receives at least k linearly independent coded symbols. The decoding phase can be done using Gaussian elimination for solving a system

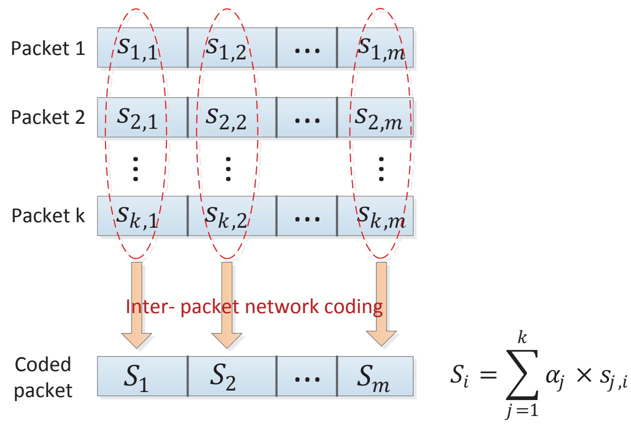


Figure 7: Inter-packet network coding.

of linear equations. Consequently, the gain from the i-th symbols of the k packets can be calculated using the following equation:

$$u_i^{NC} = w_i \times k \times \sum_{l=1}^{n} \Big[\sum_{j=k}^{x_i \times k} \binom{k \times x_i}{j} \times (1-p_l)^j \times p_l^{x_i \times k - j} \Big] \tag{5}$$

In Equation (5), we multiply w_i by k since, when we code the i-th symbols of the k packets together, any destination node can decode all of the symbols, or none of them. The total number of transmissions for the set of i-th symbols is equal to $x_i \times k$; as a result, the probability of receiving j coded symbols correctly, and happening error in the rest of the coded symbols, is equal to $\binom{k \times x_i}{j} \times (1-p)^j \times p^{x_i \times k - j}$, where $\binom{k \times x_i}{j}$ is the number of possible ways to select j coded symbols out of the transmitted coded symbols. A node needs at least k coded symbols to decode the coded symbols; therefore, the number of received coded symbols should be in the range of k and $x_i \times k$.

Because of using network coding, each coded symbol contributes the same amount of information to the destination nodes. Therefore, receiving any k coded symbols is sufficient for retrieving the symbols. This is in contrast to the case of non-coding transmissions, in which a destination node might not receive some of the symbols, and might receive the other symbols multiple times. In this case, receiving a symbol multiple times does not contribute to the total gain. However, network coding decreases the probability of receiving partial i-th symbols of the packets. The reason is that, if a destination node receives enough coded symbols, it can decode the coded packets and retrieve all of the original symbols; however, it cannot decode the coded symbols in the case of receiving an insufficient number of coded packets.

Consider the example in Figure 8, in which the source node wants to send two single symbol packets to the destination node d_1. Assume that the transmission

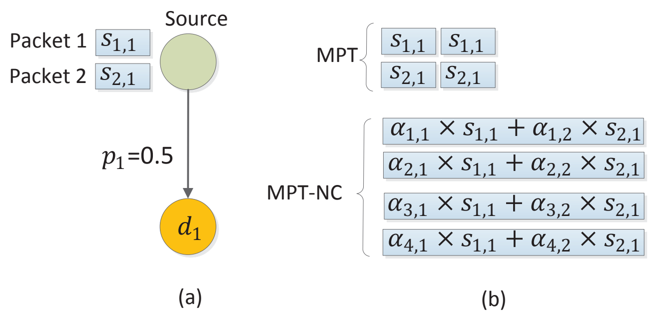


Figure 8: Example of inter-packet network coding.

error rate is equal to 0.5, and $x_1 = 2$. The WMPR scheme sends each symbol twice. As a result, the probability of the destination node receiving both of the packets is equal to $(1 - p^2) \times (1 - p^2) = 0.5625$, and the probability of receiving just one of the packets is equal to $2 \times (1 - p^2) \times p^2 = 0.3750$. On the other hand, the WMPR-NC scheme sends 4 random linear combinations of the symbols. Therefore, the destination node can decode and recover both of the symbols, if it receives at least any 2 coded symbols out of the 4 transmitted coded symbols. In this case, the probability of retrieving both of the packets is equal to $1 - p^4 - 3 \times p^3 \times (1 - p) = 0.75$, which is more than the WMPR mechanism. The reason for this difference is that, in the case of non-coded symbols, the destination node needs to receive each of the transmitted symbols at least once, and receiving one of the symbols twice does not have any advantage. However, in the case of network coding, the probability of retrieving just one of the symbols is equal to 0; as in random linear network coding, partial retrieval is not possible.

Figure 9 shows the gain of the network coding and no coding approaches for a different number of transmissions t. The number of packets and the link's error rate are equal to 10 and 0.5, respectively. It can be inferred from the figure that, in this case, for a t greater than 2, it is more efficient to use the proposed inter-packet network coding. In contrast, for a t less than or equal to 2, we should avoid using network coding, since it reduces the gain.

Referring to our discussion, for each set of symbols from the different packets, it might be beneficial to use network coding, or it might be more efficient to avoid using network coding. Therefore, for each set of the i-th symbols of the packets, we compute the utility of the non-coding and coding mechanisms. If the performance of the coding policy is more than that of the non-coding, we generate $k \times x_i$ random coded symbols, where x_i is the optimal number of transmissions when we use the non-coding mechanism. This process is shown in Algorithm 2. It should be noted that, if it is more efficient to transmit the i-th symbols of the packets without using network coding, we do not need to continue the algorithm for the remaining symbols. The reason is that, always,

Algorithm 2 WMPR-NC Algorithm

Compute the optimal $\vec{x}$ by running WRH or WR merhods
for i=1 to m **do**
  $u_i = w_i \times k \times \sum_{l=1}^{n}(1 - p_l^{x_i})$
  $u_i^{NC} = w_i \times k \times \sum_{l=1}^{n}\left[\sum_{j=k}^{x_i \times k}\binom{k \times x_i}{j} \times (1-p_l)^j \times p_l^{x_i \times k - j}\right]$
  if $u_i^{NC} > u_i$ **then**
    for i=1 to $k \times x_i$ **do**
      Create a random linear combination of the i-th symbols

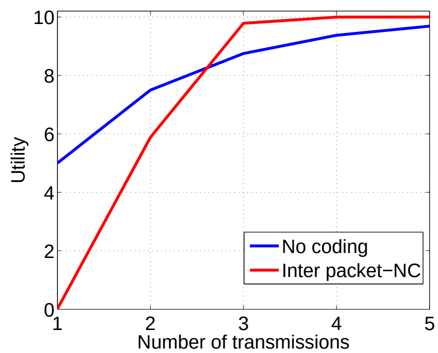


Figure 9: Comparison between the gain of the inter-packet network coding and no coding mechanisms, error probability $p = 0.5$, number of packets $k = 10$.

$x_j \le x_i, \forall i, j : j > i$, as $w_j \le w_i$. Therefore, if is not efficient to encode the i-th symbols together, it is definitely not efficient to encode the j-th symbols.

6. Implementation

6.1. Packet Header

After assigning the transmissions to the symbols, we should put them together to form the packets. In the WRH, WR, and WMPR mechanisms, we need to specify the index of each symbol in the packet. If we had just one transmission for each symbol, we could simply mention the first and the last index of the symbols that are included in the packet. Then we could put the symbols in the packet in increasing order of their index. However, in our schemes, each symbol might be included in a packet several times. As a result, we need 3 fields in the header to indicate the locations of symbol s_i. The first field represents the index of the symbol. The second and the third fields are used to show the starting and the ending locations of symbol s_i in the packet, respectively. Figure 10 (a) shows the structure of the header in the WRH, WR, and WMPR mechanisms.

The header contains important information about the location of the symbols in the packet. As a result, the header must be received correctly by the

Algorithm 3 Optimal header duplication

$Max\ gain = 0$
for $x_0 = 1$ to $t-1$ **do**
  depending on the setting, run the WRH, WR, and WMPR algorithms to compute the optimal $\vec{x}$ in transmitting $t - x_0$ symbols
  use Equation (6) to compute u
  if $u > Max\ gain$ **then**
    $Max\ gain = u$
  else
    return x_o and $\vec{x}$
    exit loop

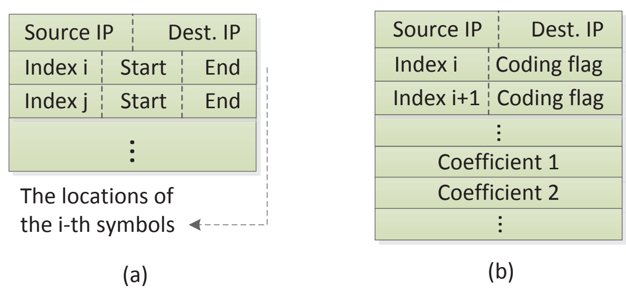


Figure 10: Packets' header, (a): The WRH, WR, and WMPR mechanisms, (b): The WMPR-NC mechanism.

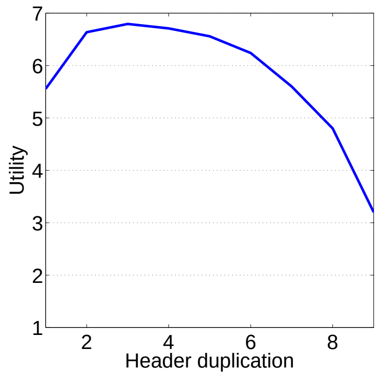
destination nodes. To increase the reliability, forward error correction (FEC) codes [31–33] can be used. In addition to FEC codes, we can include the header multiple times in the packet, as this part of the packet is much more important than the other parts. If we consider the correct delivery of the header, the Objective Function (1) can be rewritten as follows:

$$u = k \times \sum_{i=1}^{m} \sum_{l=1}^{n} w_i \times (1 - p_l^{x_o}) \times (1 - {p_l}^{x_i}) \qquad (6)$$

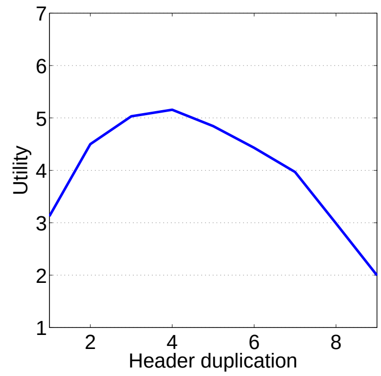
$$s.t. \quad \sum_{i=0}^{m} x_i = t$$

where x_0 is the header duplication.

Consider Figures 11 (a) and (b). We assign different values to x_0 and run the WRH algorithm to find the optimal distribution of the remaining transmissions to the symbols. Figures 11 (a) and (b) show the maximum achievable gain when the total number of transmissions is equal to 10, and the error rates are equal to 0.2 and 0.5, respectively. These figures show that, as we increase the duplication



(a)



(b)

Figure 11: Optimal header duplication, total number of transmissions equal to 10; (a): p=0.2. (b): p=0.5.

of the header, the total gain increases. The reason is that, a correctly received symbol is not useful unless the header is also received correctly. However, after a specific point, the total gain starts to decrease. To find the optimal header duplication x_0, we start with $x_0 = 1$, and run the WRH, WR, and WMPR algorithms to compute the optimal x_i in transmitting $t-1$ symbols. We repeat the same process for $x_0 = 2$ and $t-2$, and stop once we find that the utility decreases as we increment x_0. The details are shown in Algorithm 3.

In the WMPR-NC mechanism, the i-th symbol might be encoded or non-coded. Therefore, we need a flag field to indicate the encoded symbols. The packets' header in the WMPR-NC method is shown in Figure 10 (b). In addition to the source and destination IP addresses, we use *index* and *coding flag* to show the encoded symbols. The coefficients of the coded symbols are also included at the end of the header, which increases the overhead. In order to decrease this overhead, we can put some predefined random coefficient vectors on the destination and the source nodes. In this way, instead of including the coefficient in the header, the source can just put the index of the coefficient vectors in the header. In order to make the coefficient vectors useful for any packet batch size, the size of the predefined vectors should be chosen long enough. If the size of a given batch is less than the vector size, the extra elements of the vector can be ignored by the destination nodes.

6.2. Packet Header Overhead

It is clear that the header overhead of network coded packets is more than that of the uncoded packets. As a result, depending on the header costs, network coding might be efficient or inefficient. In order to consider the packet's header cost, we modify Algorithm 2 to Algorithm 4, and refer it as the WMPR-header algorithm. We denote the total gain in the case that network coding is enabled

Algorithm 4 MPT-header

$u^{NC} = 0$
$u^{UC} = 0$
Compute the optimal $\vec{x}$ by running WRH or WR methods
for i=1 to m **do**
  $u_i = w_i \times k \times \sum_{l=1}^{n}(1 - p_l^{x_i})$
  $u_i^{NC} = w_i \times k \times \sum_{l=1}^{n} \left[\sum_{j=k}^{x_i \times k} \binom{k \times x_i}{j} \times (1 - p_l)^j \times p_l^{x_i \times k - j} \right]$
  $u^{NC} = u^{NC} + \max(u_i^{NC}, u_i)$
  $u^{UC} = u^{UC} + u_i$
$u^{NC} = u^{NC} - c^{NC} \times (\sum_{i=1}^{m} w_i) \times k \times \frac{t}{m} \times n$
$u^{UC} = u^{UC} - c^{UC} \times (\sum_{i=1}^{m} w_i) \times k \times \frac{t}{m} \times n$
if $u^{NC} \le u^{UC}$ **then**
  Turn off network coding

as u^{NC}. Moreover, the total gain without network coding (uncoded packets) is represented as u^{UC}. We first compute the utilities in the cases that network coding is enabled or disabled (raw utilities). Each iteration of the loop computes the utility of each symbol. In each iteration of the for loop, we add $\max(u_i^{NC}, u_i)$ to u^{NC}, since the i-th symbols in the network coding mode can be coded or uncoded (see Algorithm 2).

After computing the raw utilities, we subtract the header costs from the raw utilities. Assume that the header cost of a linear coded packet and an uncoded packet are equal to c^{NC} and c^{UC}, respectively. The value of each packet is equal to $\sum_{i=1}^{m} w_i$. Moreover, we have k packets and each of them will be transmitted $\frac{t}{m}$ times (note that t is the total number of symbol transmissions for each packet). Consequently, the total overhead of network coded packets is equal to $c^{NC} \times (\sum_{i=1}^{m} w_i) \times k \times \frac{t}{m}$. We are computing the total utility of n nodes; thus, we multiply the overhead by n and subtract it from the raw utilities. If $u^{NC} \le u^{UC}$, we disable network coding, as it reduces the gain. Algorithm 4 shows the details.

6.3. Unknown Channel

So far, we have assumed that the channel erasure probabilities are perfectly known by the source node. The total gain is highly dependent on the error rate of the links; therefore, the source node needs to learn them, when it does not have perfect channel knowledge. For this purpose, each destination node d_i sends a feedback message to the destination node at the end of the $t \times k$ transmissions by the destination node (t transmissions for the case of single packet), which contains the number of successfully received symbols. Assume that the number of correctly received symbols in the last transmission window τ, and the estimated error rate of the destination node d_i after the τ-th set of transmissions, are equal to $r_{i,\tau}$ and $\hat{p}_{i,\tau}$, respectively. Accordingly, the estimated

Algorithm 5 Updating channels' error rate

After the τ-th set of transmissions update the error rates $p_{i,\tau+1}$, as follows,
for i=1 to n **do**
  Receive $r_{i,\tau}$ from destination d_i
  $p_{i,\tau} = \frac{t \times k - r_{i,\tau}}{t \times k}$
  $\hat{p}_{i,\tau+1} = \frac{(\tau-1)\times \hat{p}_{i,\tau} + p_{i,\tau}}{\tau}$

channel error rate of the destination node d_i is given by:

$$\hat{p}_{i,\tau+1} = \frac{(\tau - 1) \times \hat{p}_{i,\tau} + p_{i,\tau}}{\tau} \tag{7}$$

where $p_{i,\tau}$ represents the error rate of the link between the source and node d_i in the τ-th set of transmissions, and can be calculated as follows:

$$p_{i,\tau} = \frac{t \times k - r_{i,\tau}}{t \times k}$$

In Equation (7), we multiply $\tau - 1$ by $\hat{p}_{i,\tau}$ to compute the total error rate in the $\tau - 1$ set of transmissions. Then, we sum it up with the measured error rate in the last set of transmissions, and compute the average error rate. Algorithm 5 shows the updating process of the error rates.

7. Simulation

7.1. Setting

In this section, we evaluate our proposed mechanisms WRH (weighted retransmission with homogenous destinations), WR (weighted retransmission with heterogenous destinations), WMPR (weighted multiple packets retransmission), and WMPR-NC (weighted multiple packets retransmission with network coding). We compare our proposed mechanisms with a simple retransmission (SR) method. In this method, we distribute the transmissions evenly to the different symbols of the packets. As mentioned in the setting, the packets of a batch have a deadline to be received by the destination nodes, which is equal to the window size, and after this time another batch of packets will be ready for transmission. Thus, channel coding, hierarchical coding, and unequal error protection methods cannot be applied in our setting. That is the reason we do not include them in our simulations. Moreover, the objective of the mentioned papers in the related work is to provide 100% reliability, and they do not have any constraint on the number of transmissions. In contrast, we want to maximize the gain with a fixed number of transmissions. We run the simulations on 1,000 random topologies, with different links' error rates, and for each of the random topologies, we run the simulations 10 times. The plots in this paper are based on the average outputs of the simulation runs. We assume that the weight of the i-th symbol of a packet is equal to 2^{m-i}. The tunable metrics in the simulations are as follows:

- Total number of transmissions: in order to study the effect of the number of transmissions on the total gain, we evaluate the methods with a number of transmissions in the range of m and $4 \times m$ for each packet.
- Packet size: the number of symbols in each packet in different plots are in the range of 5 to 10.
- Number of packets: in the case of multiple packets transmission, we change the number of packets that the source node transmits to the destination nodes from 20 to 50.

We choose these ranges since we believe that they are reasonable numbers in a typical scenario.

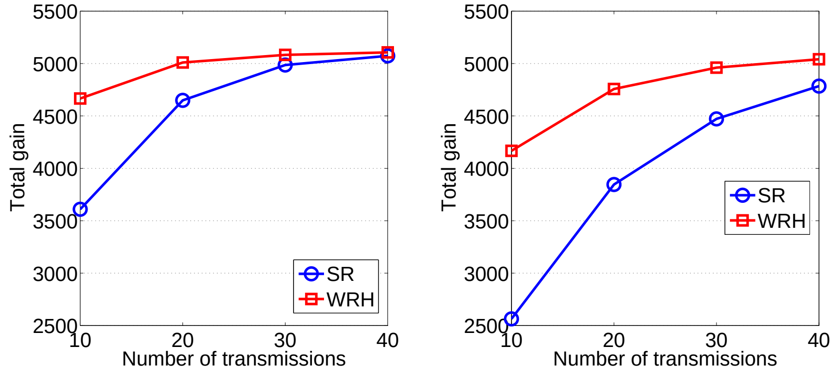
7.2. Results

7.2.1. Single Packet

in the first experiment, we compare the total gain of the WRH and the SR methods in Figure 12 (a). The packet size in this experiment is equal to 10 symbols. Also, the number of destination nodes and the link error probability are equal to 5 and 0.3, respectively. It is clear that the total gain should increase as we increase the total number of transmissions, which can be seen in the figure. Moreover, the figure shows that the difference between the WRH and the SR methods decreases as we increase the total number of transmissions from 10 to 40 symbols. The reason is that the successful delivery of all of the symbols approaches 1 in both of the mechanisms as we increase the number of retransmissions. Figure 12 (a) shows that the total gain of the WRH mechanism is up to 30% more than that of the SR method.

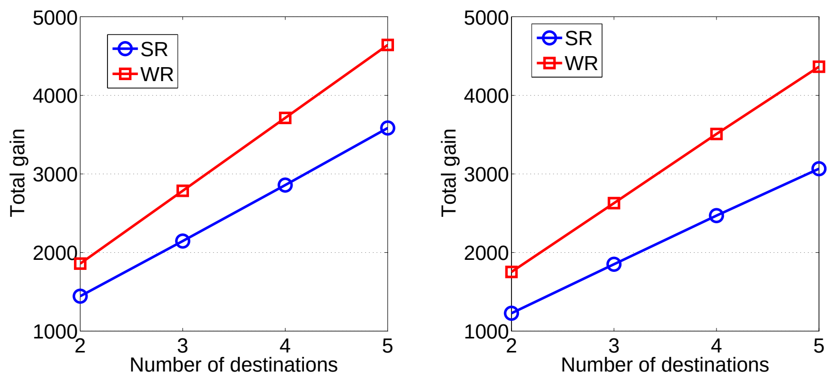
We increase the link's error rate to 0.5, and repeat the previous experiment in Figure 12 (b). Similar to Figure 12 (a), the difference between the two mechanisms decreases as we increase the number of retransmissions in Figure 12 (b). However, by comparing Figures 12 (a) and (b), we find that the efficiency of our proposed mechanism, WRH, increases as the link's error rate increases. The total gain of the WRH approach in this figure is up to 60% more than that of the SR method.

In the next experiment, we evaluate the gain of the WR mechanism in sending a packet to multiple destinations, by comparing it to the SR method in Figure 13 (a). We set the packet size to 10 symbols, and transmit a total of 10 symbols. In each of the 1,000 runs, the links' error rates are randomly chosen in the range of $[0.2, 0.4]$. The figure shows that the gain of both of the mechanisms increase as we increase the number of destinations; this is due to the presence of more receiver nodes. Also, it is clear from the figure that the relationship of the total gain and the number of destinations is linear, which is because of the independence of the links. As a result, the ratio of the gain of the mechanisms is fixed in this figure.



(a) (b)

Figure 12: Comparison between the gain of the WRH and SR mechanisms in the case of single packet transmission, $m = 10$, $k = 1$, $n = 5$; (a) $p = 0.3$, (b) $p = 0.5$.



(a) (b)

Figure 13: Comparison between the gain of the WR and SR mechanisms in the case of single packet transmission, $m = 10$, $k = 1$, $t = 10$; (a) $p \in [0.2, 0.4]$, (b) $p \in [0.2, 0.6]$.

We repeat the previous experiment in Figure 13 (b) by increasing the range of the links' error rates to $[0.2, 0.6]$. As it is expected, the gains of the mechanisms in Figure 13 (b) are less than that of Figure 13 (a). The efficiency of the WR mechanism increases as the error rates increase.

7.2.2. Multiple Packets

Figure 14 (a) shows the total gain of the WMPR, WMPR-NC, and SR mechanisms. In this figure, the packet size is equal to 5 symbols. Also, the number of destination nodes is equal to 5, and the error rate of the links between

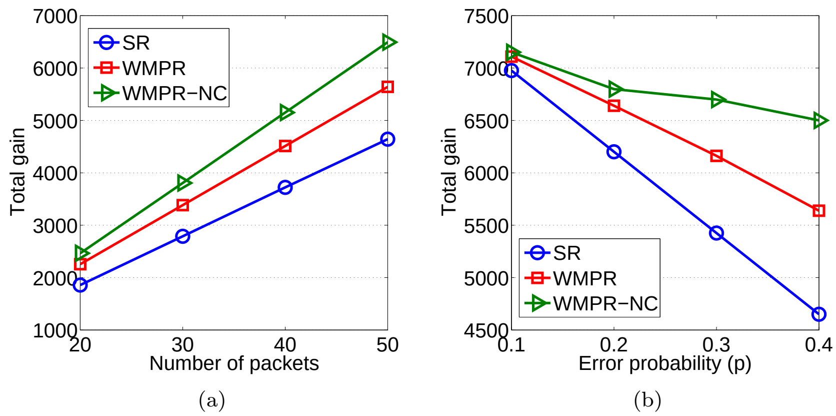


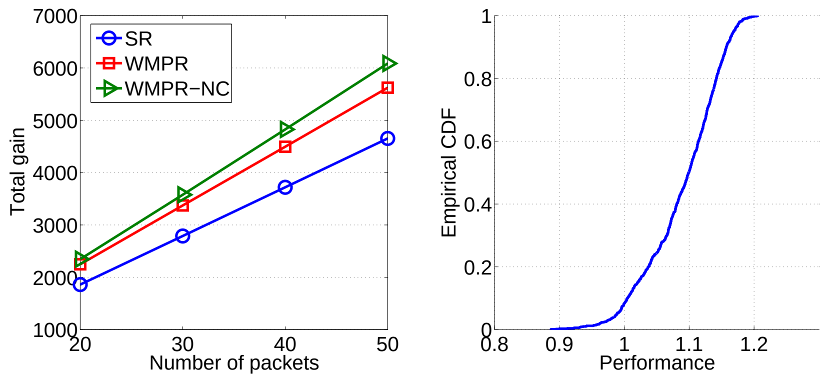
Figure 14: Comparison between the gain of WMPR, WMPR-NC, and SR mechanisms, $m = 5$, $n = 5$; (a) $p = 0.4$, $t = 5$ (b) $k = 50$.

the source and the destination node is equal to 0.4. We increase the total number of transmissions as we increase the number of packets, and it is equal to the total number of symbols (total number of symbols is equal to 5 times the number of packets). As it is expected, the gain of the WMPR-NC mechanism is more than that of the other methods. Moreover, the gain of the WMPR mechanism is more than that of the SR method. Figure 14 (a) shows that the gain of the WMPR-NC mechanism is up to 15%, and 45% more than that of the WMPR and SR methods, respectively. Also, the efficiency of the network coding increases as we increase the number of packets, which are coded together.

We evaluate the effect of the link's error rate on the gain in Figure 14 (b). The packet size and the number of packets are equal to 5 symbols and 50, respectively. Also, for the total 250 symbols that the source node needs to transmit, we set the total number of transmissions to 250. The figure shows that the total gain of the WMPR and SR mechanisms drop dramatically as we increase the error rate. In contrast with the other methods, WMPR-NC is more robust to the error rate, which is due to the use of network coding.

We repeat the experiment of Figure 14 (a) in Figure 15 (a) with 5 destination nodes. The packet size is equal to 5 symbols, and the links' error rates are in the range of $[0.3, 0.5]$. Much similar to Figure 14 (a), the gain of all of the mechanisms increase as we increase the number of packets. Note that we increase the total number of transmissions as we increase the number of packets. By comparing Figure 14 (a) with Figure 15 (a), we find that the difference between the WMPR and WMPR-NC decreases in the case of multiple destinations, which is because of the diversity of the links. Consequently, the efficiency of WMPR-NC increases in the case that the error rates of the links are close to each other.

We compare the performance of the WMPR-NC mechanism to the WMPR in Figure 15 (b). For this purpose, we divide the gain of the WMPR-NC mech-



(a) (b)

Figure 15: Comparison between the gain of WMPR, WMPR-NC, and SR mechanisms, $m = 5$, $n = 5$, $p \in [0.3, 0.5]$; (a) total gain, $t = 5$ (b) Performance of the WMPR-NC mechanism over the WMPR method.

anism by that of the WMPR mechanism, and plot its CDF. In this experiment, the packet size and the number of packets are equal to 5 symbols and 50, respectively. Also, the error rate of the links between the source and the 5 destination nodes are in the range of [0.3,0.5]. This figure shows that, in less than 5% of the cases, the number of delivered symbols in the WMPR-NC mechanism is less than that of the WMPR method. Moreover, in more than 50% of the cases, the number of delivered symbols of the WMPR-NC protocol is more than 10% higher than that of the WMPR mechanism.

Figure 16 (a) shows the gain of the WMPR, WMPR-NC, and WMPR-header. We set the header cost of the coded and uncoded packets to 0.07 and 0.05, respectively. The number of packets and symbols in each packet are equal to 20 and 5. Moreover, the size of the transmission time window for each packet is set to 5. Figure 16 (a) shows that, for more reliable links, performing network coding might not be efficient, as the gain of WMPR-NC is less than that of the WMPR method. The reason is that, for these cases, the advantage of performing network coding over uncoding is less than the increase in the overhead. The WMPR-header considers the header overhead of the packets; as a result, it disables network coding when it finds that coding is not efficient. As we increase the error rate of the links, the difference between the utility of network coding and uncoding increases. Therefore, the utility of WMPR-NC becomes more than that of the WMPR method, and the WMPR-header method automatically switches to coding.

We increase the header cost of the coded packet c^{NC} to 0.09, and repeat the previous experiment. Figure 16 (b) shows the simulation result. Increasing c^{NC} reduces the utility of network coding; thus, the WMPR-header method turns on coding in the case of less reliable links. Note that in this simulation

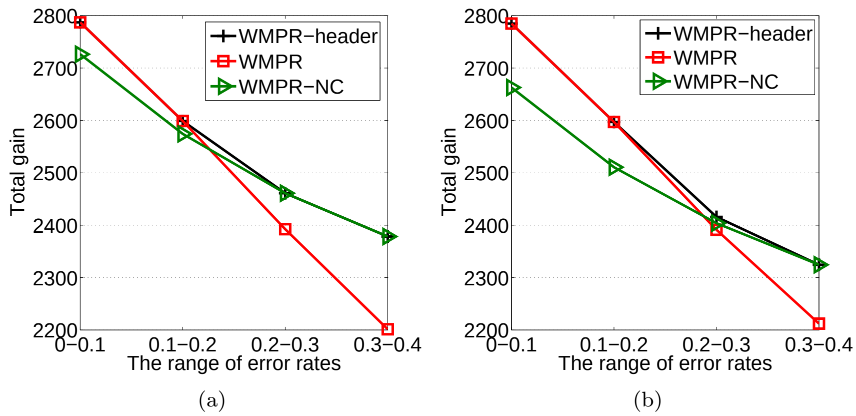


Figure 16: Comparison between the gain of WMPR, WMPR-NC, and WMPR-header mechanisms, $m = 5$, $n = 5$, $k = 20$, $t = 5$; (a) $c^{NC} = 0.07$, $c^{UC} = 0.05$ (b) $c^{NC} = 0.09$, $c^{UC} = 0.05$.

the number of packets is fixed, and an alternative way to make network coding more efficient is increasing the number of packets k, as shown in Figure 15 (a).

8. Conclusion

There is much work on reliable transmissions over error-prone wireless channels. In contrast to the previous work on reliable transmission, we consider a novel problem in this paper. We study the problem of maximizing the total gain in the case of partial data delivery in error-prone wireless networks. In our setting, each set of bits, called symbols, has a different weight. We first address the case of single packet transmission to a homogenous destination nodes, and we show that the optimal solution of this problem has a round-robin pattern. Then, we extend our solution to the case of heterogenous destinations. We also provide a solution for the case of sending multiple packets to multiple destinations, and we enhance the expected gain (utility) using inter-packet random linear network coding.

Our extensive results show that our proposed multiple packets transmission mechanism can increase the gain up to 60%, compared to that of a simple retransmission mechanism. Moreover, using random linear network coding can enhance the gain.

acknowledgment

This research was supported in part by the NSF grants ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

[1] H. Djandji, An efficient hybrid arq protocol for point-to-multipoint communication and its throughput performance, IEEE Transactions on Vehicular Technology 48 (5) (1999) 1688–1698.

[2] B. Zhao, M. Valenti, The throughput of hybrid-ARQ protocols for the gaussian collision channel, IEEE Transactions on Information Theory 47 (5) (2001) 1971–1988.

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

[4] A. Shokrollahi, Raptor codes, IEEE Transactions on Information Theory 52 (6) (2006) 2551–2567.

[5] D. J. MacKay, Fountain codes, IEE Proceedings- Communications 152 (6) (2005) 1062–1068.

[6] T. Cover, J. Thomas, Elements of information theory, Wiley-InterScience, 2006.

[7] S. McCanne, V. Jacobson, M. Vetterli, Receiver-driven layered multicast, in: ACM CCR, 1996, pp. 117–130.

[8] N. Shacham, Multipoint communication by hierarchically encoded data, in: IEEE INFOCOM, 1992, pp. 2107–2114.

[9] M. Effros, Universal multiresolution source codes, IEEE Transactions on Information Theory 47 (6) (2001) 2113–2129.

[10] L. Rizzo, L. Vicisano, RMDP: an FEC-based reliable multicast protocol for wireless environments, ACM SIGMOBILE Mobile Computing and Communications Review 2 (2) (1998) 23–31.

[11] L. Rizzo, Effective erasure codes for reliable computer communication protocols, ACM SIGCOMM Computer Communication Review 27 (2) (1997) 24–36.

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

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, Xors in the air: practical wireless network coding, in: ACM SIGCOMM, 2006.

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

[15] S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading structure for randomness in wireless opportunistic routing, in: ACM SIGCOMM, 2007.

[16] P. Ostovari, J. Wu, A. Khreishah, Network coding techniques for wireless and sensor networks, in: H. M. Ammari (Ed.), The Art of Wireless Sensor Networks, Springer, 2013.

[17] P. Ostovari, J. Wu, A. Khreishah, Deadline-aware broadcasting in wireless networks with local network coding, in: IEEE ICNC, 2012.

[18] P. Ostovari, A. Khreishah, J. Wu, Deadline-aware broadcasting in wireless networks with network coding, in: IEEE GLOBECOM, 2012.

[19] A. Khreishah, I. Khalil, P. Ostovari, J. Wu, Flow-based xor network coding for lossy wireless networks, IEEE Transactions on Wireless Communications 11 (6) (2012) 2321–2329.

[20] R. Ahlswede, N. Cai, S. Li, R. Yeung, Network information flow, IEEE Transactions on Information Theory 46 (4) (2000) 1204–1216.

[21] S. Li, R. Yeung, N. Cai, Linear network coding, IEEE Transactions on Information Theory 49 (2) (2003) 371–381.

[22] R. Koetter, M. Medard, An algebraic approach to network coding, IEEE/ACM Transactions on Networking 11 (5) (2003) 782– 795.

[23] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, IEEE Transactions on Information Theory 52 (10) (2006) 4413–4430.

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

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

[26] D. Nguyen, T. Tran, T. Nguyen, B. Bose, Wireless broadcast using network coding, IEEE Transactions on Vehicular Technology 58 (2) (2009) 914–925.

[27] W. Fang, F. Liu, Z. Liu, L. Shu, S. Nishio, Reliable broadcast transmission in wireless networks based on network coding, in: Computer Communications Workshops (INFOCOM WKSHPS), 2011, pp. 555–559.

[28] S. Katti, D. Katabi, H. Balakrishnan, M. Medard, Symbol-level network coding for wireless mesh networks, ACM SIGCOMM Computer Communication Review 38 (4) (2008) 401–412.

[29] M. Li, Z. Yang, W. Lou, Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding, IEEE Journal on Selected Areas in Communications 29 (1) (2011) 223–235.

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

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

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

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

Appendix A. Optimality of WRH Method

Here, we prove the optimality of the WRH mechanism, and we show that the optimal solution has a round-robin pattern. The utility function in the case of transmitting one packet to homogeneous destinations is as follows:

$$\begin{aligned} u &= \sum_{i=1}^{m} n \times w_i \times (1 - p^{x_i}) \\ s.t. \quad & \sum_{i=1}^{m} x_i = t \end{aligned}$$

For the packet size equal to 2 symbols ($m = 2$) we have:

$$\begin{aligned} u &= n \times \left[w_1 \times (1 - p^{x_1}) + w_2 (1 - p^{x_2}) \right] \\ s.t. \quad & x_1 + x_2 = t \end{aligned}$$

Appendix A.1. Proof of Optimality for the Case m=2

Lemma 1. *If* $p^{x_1} < \frac{w_1}{w_2} p^{x_2}$*, then* $p^{x_1} > \frac{w_1}{w_2} p^{x_2+1}$.

Proof. We use contradiction to proof Lemma 1. We refer to the optimal solution at the current iteration as (x_1, x_2). Assume that the current state is (x_1, x_2) and $p^{x_1} < \frac{w_1}{w_2} p^{x_2+1}$. As a result, $p^{x_1-1} < \frac{w_1}{w_2} p^{x_2}$, and we have:

$$w_1 p^{x_1-1} < w_2 p^{x_2}$$

By multiplying the two sides of this inequality with $n \times (1 - p)$ we will have:

$$\begin{aligned} & n \times w_1 \times (1-p) p^{x_1-1} < n \times w_2 \times (1-p) p^{x_2} \\ & \Rightarrow \Delta_{x_1-1} < \Delta_{x_2} \end{aligned}$$

As a result, it should be more efficient to increase x_2 in the previous iteration. Therefore, in the current iteration we will have (x_1-1, x_2+1), which contradicts the assumption that the current state is (x_1, x_2). Consequently, we have $p^{x_1} > \frac{w_1}{w_2} p^{x_2+1}$. ■

Lemma 2. *If* $p^{x_1} > \frac{w_1}{w_2} p^{x_2}$*, then* $p^{x_1+1} < \frac{w_1}{w_2} p^{x_2}$.

Proof. We use contradiction to proof Lemma 1. Assume that the current state is (x_1, x_2) and $p^{x_1+1} > \frac{w_1}{w_2} p^{x_2}$. As a result, $p^{x_1} > \frac{w_1}{w_2} p^{x_2-1}$, so we have:

$$w_1 p^{x_1} > w_2 p^{x_2-1}$$

By multiplying the two sides of this inequality with $1-p$ we will have:

$$\begin{aligned}&n \times w_1 \times (1-p)p^{x_1} > n \times w_2 \times (1-p)p^{x_2-1}\\&\Rightarrow \Delta_{x_1} > \Delta_{x_2-1}\end{aligned}$$

Therefore, it should be more efficient to increment x_2 in the previous state. Thus, in the current state, we will have (x_1+1, x_2-1), in which $x_2 \geq 1$ ($x_2 - 1$ cannot be negative) contradicts the assumption that the current state is (x_1, x_2). Consequently, $p^{x_1+1} < \frac{w_1}{w_2} p^{x_2}$. ■

Proposition 1. *Assigning the transmissions to x_1 for $x_1 \leq \log_p \frac{w_2}{w_1}$ and then incrementing x_1 and x_2 in a round-robin pattern will result in the optimal solution.*

Proof. Based on Equation 3, if $p^{x_1} < \frac{w_2}{w_1}$ then $\Delta_{x_1} < \Delta_{x_2}$, so x_2 should be zero. In addition, based on Lemma 1 after this point, every time we increment x_2, Δ_{x_2+1} becomes less than Δ_{x_1}. Therefore, in this case, assigning the next transmission to x_1 results in a larger gain. Lemma 2 is the reverse of Lemma 1, which results in a round-robin incrementing pattern. ■

Appendix A.2. Proof of Optimality for the case general m

Lemma 3. *If $p^{x_i} > \frac{w_j}{w_i} p^{x_j} \forall i, j \in [1, m], j \neq i$, then $p^{x_i+1} < \frac{w_j}{w_i} p^{x_j}$.*

Proof. Assume that the current state is $(x_1, x_2, ..., x_m)$, and there is a j such that $p^{x_i+1} > \frac{w_j}{w_i} p^{x_j}$. Then, $p^{x_i} > \frac{w_j}{w_i} p^{x_j-1}$ in one of the previous states. As a result, $\Delta_{x_i} > \Delta_{x_j-1}$, so we should see a state with $x_i + 1$ and $x_j - 1$. In this case, there is no way to see the current state, which contains x_i and x_j. ■

Proposition 2. *The WRH algorithm results in an optimal solution.*

Proof. It can be inferred from Lemma 3 that the optimal assignment has a round-robin pattern. The reason is that, when we increment x_i, p^{x_i} becomes less than $\frac{w_j}{w_i} p^{x_j}, \forall j : j \neq i$. The next time p^{x_i} becomes greater than $\frac{w_j}{w_i} p^{x_j}$ is when we increment all $x_j, j \neq i$. ■

Appendix B. Optimality of the WR Method

Lemma 4. *The optimal x_i, $\forall 1 \leq i \leq m$ are non-decreasing as we increase the number of transmissions t.*

Proof. The utility of a symbol s_i is equal to:

$$\sum_{l=1}^{n} w_i \times (1 - p_l^{x_i}) \tag{B.1}$$

which is a non-decreasing function. Therefore, assigning more transmissions to a symbol results in more utility. Moreover, the utility of each symbol is a summation of concave functions; therefore that is a concave function. It means that the Δ_{x_i}, $\forall 1 \leq i \leq m$ is a decreasing function. Assume that for a given t', the optimal number of transmissions for symbols s_j and s_k are equal to x_j and x_k, respectively. Moreover, for a $t > t'$ transmissions, the optimal number of transmission for s_j and s_k are $x_j - y$ and $x_k + y$, respectively, where y is a given positive number. It contradicts with the optimality of x_j and x_k transmissions in the case of t' total transmissions. The reason is that if $x_j - y$ and $x_k + y$ results in more gain, then x_j and x_k cannot result in optimal solution for the case of t' transmissions. Note that this holds since the utility of each symbol (Equation (B.1)) is a concave and non-decreasing function. Consequently, x_i are non-decreasing. ■

The following corollary can be concluded from Lemma 4.

Corollary 1. *The optimal solution for t transmissions can be calculated from the optimal solution for $t' < t$ transmissions.*

Proposition 3. *The WR algorithm results in an optimal solution.*

Proof. We proof the optimality of the WR algorithm by induction. Let $t = 1$. It is obvious that the transmission should be assigned to the symbol s_i with the maximum Δ_{x_i}. Now, assume that for $t - 1$ transmissions the optimal solution is $(x_1, .., x_m)$. By Lemma 4, each x_i is non-decreasing. Therefore, from Corollary 1, in order to find the optimal solution for t transmissions, we just need to find the symbol s_i with the maximum Δ_{x_i} and increase x_1 by one. That is exactly the same as what the WR algorithm performs. ■