Efficient Symbol-Level Transmission in Error-Prone Wireless Networks

Pouya Ostovari*, Jie Wu*, and Abdallah Khreishah†
*Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
†Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102

***Abstract*—Providing reliable transmission over error-prone networks has received a lot of attention from the research community. In this paper, instead of using simple retransmissions to provide reliability, we consider a novel retransmission approach based on the importance of the 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 (symbols) has a different weight. We first address the case of one-hop single packet transmission, and prove that the optimal solution has a round-robin 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.**

***Index Terms*—Symbol-level coding, random linear network coding, weight, wireless networks, error-prone channel.**

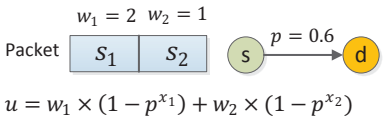
I. Introduction

Broadcasting is an essential method for disseminating data in wireless networks. However, the error-prone wireless links creates challenges. To handle these challenges, different mechanisms [1]–[4] 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 (less significant bit). Therefore, any mechanism that addresses numeric data transmissions in 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 this paper, we consider a novel broadcasting approach 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 destinations, 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 parts of the data that are more important than others.

This research was supported in part by NSF under grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

978-1-4673-5828-6/13/$31.00 ©2013 IEEE



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

(a)

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

(b)

Fig. 1. Motivation example; (a) 2 transmissions, (b) 3 transmissions.

Consider the example in Fig. 1, in which a packet is transmitted to a destination. The weights of the symbols s_1 and s_2 is equal to 2 and 1, respectively. Assume that the error-rate (p) 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 the transmissions and their respective utilities are shown in Fig. 1(a). 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. Fig. 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 [5].

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

- In contrast to previous works, which study the problem of reliable packets 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 and single destination, 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 algorithm for the case of transmitting a single packet to multiple destinations.
- We extend the proposed single packet transmission algorithms to the case of multiple packets, and use the advantage of random linear network coding (NC) to enhance the expected gain. We show that NC does not necessarily increase the gain, and we find the condition that NC results in more gain than the non-coding mechanism.

The rest of this paper is organized as follows. Section II reviews the related work. In Section III, we provide the problem definition and the setting. We propose our mechanisms for the case of transmitting a single packet in Section IV. In Section V, we extend our proposed mechanism to the case of transmitting multiple packets, and we boost the gain using linear inter-packets NC. We discuss the implementation issues in Section VI, and evaluate the proposed mechanisms through simulations in Section VII. Section VIII concludes the paper.

II. Related Work and Background

A. Reliable Transmission

Certain mechanisms, such as feedback messages, are used in 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 a lot of feedback messages, especially for the case of multi destination nodes. Hybrid-ARQ approaches [2], [6], which combine FEC (Forward Error Correction) with ARQ, are proposed to solve this problem.

An efficient way to provide reliability without using feedback is to use rateless (fountain) codes [3], [4]. In these schemes, the source node can generate and transmit an unlimited number of encoded packets until every destination receives enough packets to retrieve the original packets. In rateless codes, the destination nodes need to collect a sufficient number of packets, regardless of which packets have been lost. Assuming that the number of original packets is k, the number of sufficient coded packets is $N = (1 + \epsilon)$ [3], where ϵ is a small number and shows the overhead of the rateless codes. It can be shown that as $k \to \infty$, the overhead goes to zero [7]. 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 the delay-sensitive applications, such as our problem, which needs small batches of packets.

B. Network Coding

Network coding (NC) [8]–[13] is introduced in [14] for wired networks, to solve the bottleneck problem in single multicast problem. The authors in [15] provide a useful algebraic representation of the linear NC problem. In [16], 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 NC, coded packets are random linear combinations 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 NC, the source node generates and transmits random coded packets. 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 [17]–[19] use the benefit of NC in their 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 Fig. 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, 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 NC for wireless mesh networks is introduced in [20], and it is shown that its throughput is more than that of the packet-level coding. The insight behind the symbol-level coding is that, even in the case that a node does not receive a 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, which reduces the transmission cost. The authors in [21], [22] use the symbol-level coding to propose a method for distributing data in vehicular networks.

III. Setting

Consider a single-hop wireless network that consists of one source and n destination nodes d, as depicted in Fig. 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 has a weight w_i, and in general $w_j > w_{j+1}$, $\forall j : 1 \leq j \leq m-1$. In our model, the weight of the i-th symbols of all of the packets are the same. 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.

We assume that 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. Moreover, this window size for a batch of packets is enough for transmitting $t \times k$ symbols, where t is the assigned window for a single packet. If the packets are not delay sensitive or the source has infinite

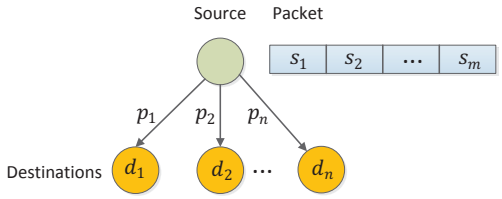


Fig. 2. System setting.

TABLE I
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)
$s_{j,j}$	The i-th symbol of the j-th packet
S_i, P_i	The i-th coded symbol, 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 function) from the i-th symbols
u_i^{NC}	The gain (utility function) from the i-th symbols when we use linear NC

packets to transmit, the optimal solution is a simple extension of the well-known channel coding theory [5]. Our goal is maximizing the total weight of the received symbols of a batch of k packets by the destination nodes. As a result, our utility function becomes:

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

It is obvious that the more important symbols should be transmitted more 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 duplications to the different symbols in order to maximize the total gain. Our goal in this work is finding this optimal assignment. The set of symbols used in this paper is summarized in Table I.

IV. Optimal Solution for the Case of Single Packet

A. One Destination

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.

1) packet size $m = 2$: For a packet size $m = 2$ the objective function becomes:

$$u = w_1 \times (1 - p^{x_1}) + w_2(1 - p^{x_2})$$
$$st. \quad x_1 + x_2 = t$$

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

$$\Delta_{x_i} = w_i \times (1 - p^{x_i+1} - (1 - p^{x_i}))$$
$$= w_i \times (1 - p) \times p^{x_i}$$

We know that $w_1 > w_2$. Therefore, it is clear 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 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 x_2. In other words, the condition to increase x_2 is $\Delta_{x_1} < \Delta_{x_2}$. Consequently, we have:

$$w_1 \times (1 - p)p^{x_1} < w_2 \times (1 - p)p^{x_2}$$
$$p^{x_1} < \frac{w_2}{w_1} p^{x_2} \quad (2)$$

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

$$p^{x_1} < \frac{w_2}{w_1} \quad (3)$$

As a result, 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.

Fig. 3 shows the optimal distribution of transmissions between x_1 and x_2 for different total numbers of transmissions t. 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 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:

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

Before 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 becomes greater than 1. As a result, the next transmission should be assigned to x_1. 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 increment x_1 until $p^{x_1} < \frac{w_2}{w_1}$. If anymore transmissions are left, we start to distribute the remaining transmissions between x_1 and x_2 in a round-robin pattern. We prove the optimality of this algorithm in the Appendix.

2) General packet size m: Similar to the case of $m = 2$, the first symbol 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

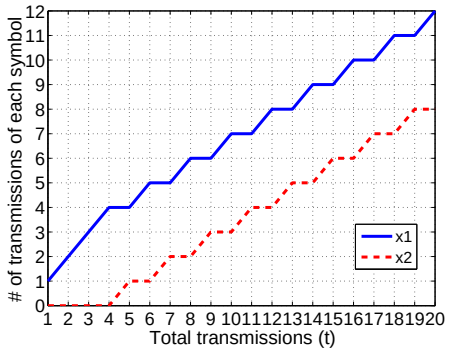


Fig. 3. Optimal distribution of transmissions between 2 symbols for an error probability $p = 0.5$, $w_5 = 5$, and $w_2 = 1$.

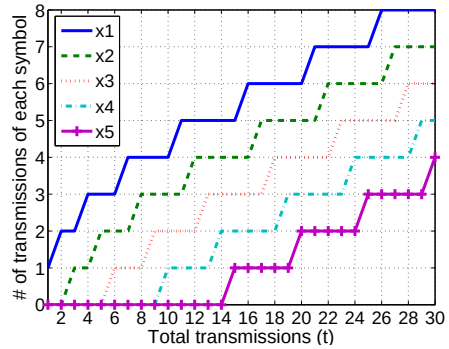


Fig. 4. Optimal distribution of transmissions between 5 symbols for an error probability $p = 0.5$, $w_i = 2^{5-i}$.

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 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:

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

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

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

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 to increase x_m is when $p^{x_i} < \frac{w_m}{w_i}, \forall i : 1 \leq i \leq m-1$. Fig. 4 shows the optimal distribution of the transmissions 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.

Algorithm 1 SPTMD 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
```

Based on the discussion, our Single Packet Transmission (SPT) algorithm works as follows. We assign the transmissions to x_1 until $p^{x_1} < \frac{w_2}{w_1}$. Then, we distribute the 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 pattern among x_1, x_2, and x_3. In general, we start incrementing x_j when $p^{x_i} < \frac{w_j}{w_i}, \forall i : 1 \leq i \leq j-1$, and we add x_j to the round-robin incrementing pattern. The proof of this algorithm's optimality is presented in the Appendix.

B. Multiple Destination Nodes

We assume that the error rate of the links are independent. As a result, in the case that the error rates of the destination nodes are equal, the problem becomes exactly the same as the case of single destination node, and we can use the proposed mechanism in the previous section. However, in the case of multiple destination nodes with different transmission error rates, the round-robin pattern does not exist. For this reason, we use an iterative algorithm, which we call it Single Packet Transmissions to Multiple Destinations (SPTMD). In the case of multiple destination nodes, Δ_{x_i} can be calculated as follows:

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

The SPTMD 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, where $j = \arg\max_{1 \leq i \leq m} \Delta_{x_i}$. 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 SPTMD method is in the order of $O(t \times m \times n)$.

V. Efficient Solution in the Case of Multiple Packets

In order to transmit a batch of k packets from a source node to a set of destination nodes, we can use two approaches: with-

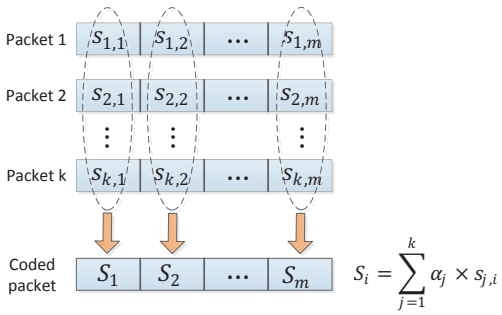


Fig. 5. Inter-packet network coding.

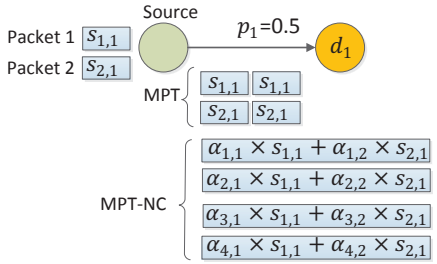


Fig. 6. Inter-packet network coding.

out and with NC. We describe the details of the mechanisms in the following sections.

A. Without Network Coding

In our model, the size of the packets (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 Multiple Packets Transmission (MPT) 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, depending on the number of destination nodes. 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.

B. Inter-Packet Network Coding

We can benefit from random linear NC in order to increase the gain of the MPT mechanism. For this purpose, much similar to the MPT method, we run the SPTMD or SPT algorithms to compute the value of optimal x_i. Then, as it is shown in Fig. 5, 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 packets. This is in contrast with the previous 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).

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

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

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 $x_i \times k$.

Using NC, 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. It is in contrast with 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, NC 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, but it cannot decode the coded symbols in the case of receiving an insufficient number of coded packets.

Consider the example in Fig. 6, in which the source node wants to send two single-symbol packets to the destination node d_1. Assume that the transmission error rate is equal to 0.5, and $x_1 = 2$. The MPT scheme sends each symbol twice. As a result, the probability of receiving both of the symbols by the destination node is equal to $(1-p^2) \times (1-p^2) = 0.5625$, and the probability of receiving just one of the symbols is equal to $2 \times (1-p^2) \times p^2 = 0.3750$. On the other hand, the MPT-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 symbols is equal to $1-p^4-3\times p^3 \times (1-p) = 0.75$, which is more than the MPT 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 NC, the probability of retrieving just one of the symbols is equal to 0, as in random linear NC, there is no way for partial retrieval.

Algorithm 2 MPT-NC Algorithm

Compute the optimal $\vec{x}$ by running Algorithm 1 (SPTMD)
for i=1 to m **do**
  $u_i = w_i \times k \times (1 - p_l^{x_i})$
  $u_i^{NC} = w_i \times k \times \sum_{l=1}^{n} \left[\sum_{j=k}^{x_i \times k} \binom{k \times x_i}{j} \times (1-p_l)^j \times p_l^{x_i \times k - j} \right]$
  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

Algorithm 3 Optimal header duplication

$Max\ gain = 0$
for $x_0 = 1$ to $t-1$ **do**
  depending on the setting, run the MPT, SPTMD, or SPT algorithms to compute the optimal $\vec{x}$ in transmitting $t - x_0$ symbols
  use Equation (5) to compute u
  if $u > Max\ gain$ **then**
    $Max\ gain = u$
  else
    return x_o and $\vec{x}$, exit loop

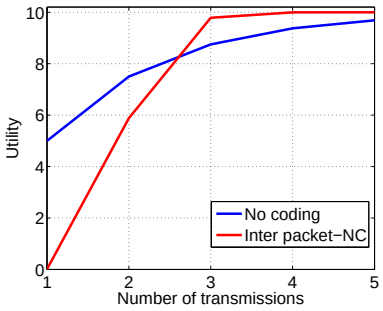


Fig. 7. Comparison between the gain of the inter-packet NC and no coding mechanisms, error probability $p = 0.5$, number of packets $k = 10$.

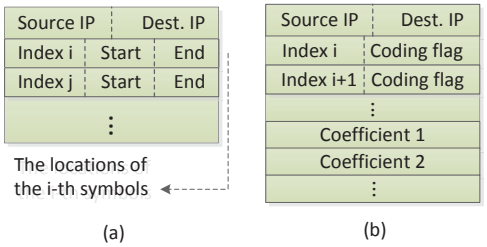


Fig. 8. Packets' header, (a): The MPT, SPTMD, and SPT mechanisms, (b): The MPT-NC mechanism.

The gain of the NC and no coding approaches for different t are shown in Fig. 7. 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 NC. In contrast, for a t less than or equal to 2, we should avoid using NC, since it reduces the gain.

Based on our discussion, for each set of symbols from the different packets, it might be beneficial to use NC, or it might be more efficient to avoid using NC. 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 NC, we do not need to continue the algorithm for the remaining symbols. The reason is that, always, $x_j \leq x_i, \forall i, j : j > i$, as $w_j \leq w_i$. Therefore, if is not efficient to encode the i-th symbols together, it is definitely not efficient to encode the j-th symbols.

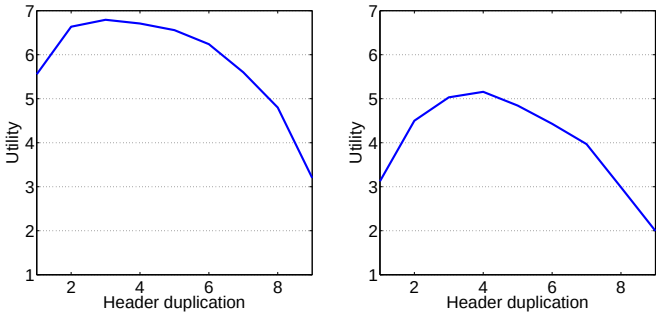
VI. Implementation

After assigning the transmissions to the symbols, we should put them together to form the packets. In the MPT, SPTMD, and SPT 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, and 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. Fig. 8(a) shows the structure of the header in the MPT, SPTMD, and SPT mechanisms.

The header must be received correctly by the destination nodes, because it contains important information about the location of the symbols in the packet. To increase the reliability, Forward Error Correction (EFC) codes can be used. In addition to EFC 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:

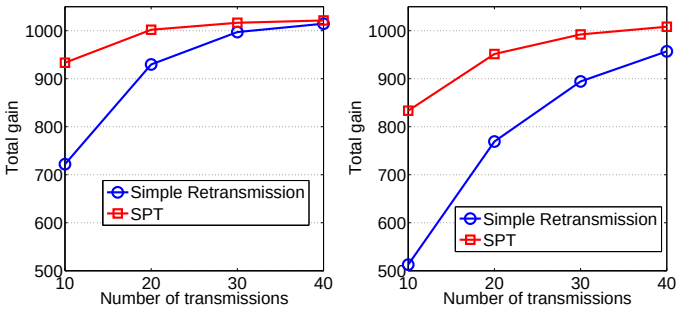
$$u = k \times \sum_{i=1}^{m} \sum_{l=1}^{n} w_i \times (1 - p_l^{x_o}) \times (1 - p_l{}^{x_i}) \qquad (5)$$
$$s.t. \quad \sum_{i=0}^{m} x_i = t$$

where x_0 is the header duplication. Consider Figs. 9(a) and (b). We assign different values to x_0 and run the SPT algorithm to find the optimal distribution of the remaining transmissions to the symbols. Figs. 9(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 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



(a) (b)

Fig. 9. Optimal header duplication, Total number of transmissions equal to 10; (a): p=0.2. (b): p=0.5.



(a) (b)

Fig. 10. Comparison between the gain of the simple retransmission and SPT mechanisms in the case of single packet transmission, $m = 10$, $n = 1$, $k = 1$; (a) $p = 0.3$, (b) $p = 0.5$.

to decrease. To find the optimal header duplication x_0, we start with $x_0 = 1$, and run the MPT, SPTMD, or SPT 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 utility decreases as we increment x_0. The details are shown in Algorithm 3.

In the MPT-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 MPT-NC method is shown in Fig. 8(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 heather, the source can just put the index of the coefficient vectors in the heather. 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 the batches is less than the vector size, the extra elements of the vector can be ignored by the destination nodes.

VII. Simulation

In this section we evaluate the SPT (Single Packet Transmission), SPTMD (Single Packet Transmission to Multiple Destinations), MPT (Multiple Packets Transmission), and MPT-NC (Multiple Packet Transmission with NC) mechanisms. We compare our proposed mechanisms with a simple retransmission method. In this method, we distribute the transmissions evenly to the different symbols of the packets. 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}.

1) Single Packet: in the first experiment, we compare the total gain of the SPT and the simple retransmission method in Fig. 10(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 1 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 SPT and the simple retransmission 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. Fig. 10(a) shows that the total gain of the SPT mechanism is up to 30% more than that of the simple retransmission method.

We increase the link's error rate to 0.5, and repeat the previous experiment in Fig. 10(b). Similar to Fig. 10(a), the difference between the two mechanisms decreases as we increase the number of retransmissions. However, by comparing Figs. 10(a) and (b) we find that the efficiency of our proposed mechanism SPT, increases as the link's error rate increases. The total gain of the SPT approach in this figure is up to 60% more than that of the simple retransmission method.

In the next experiment, we evaluate the gain of the SPTMD mechanism in sending a packet to multiple destinations, by comparing it to the simple retransmission method in Fig. 11(a). We set the packet size to 10 symbols, and totally transmit 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, which is due to 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.

We repeat the previous experiment in Fig. 11(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 Fig. 11(b) are less than that of the Fig. 11(a). The efficiency of the SPTMD mechanism increases as the error rates increase.

2) Multiple Packets: Fig. 12(a) shows the total gain of the MPT, MPT-NC, and simple retransmission mechanisms. In this figure, the packet size is equal to 5 symbols. Also, the number of destination nodes is equal to 1, and the error rate of the links between the source and the destination node is equal to 0.4.

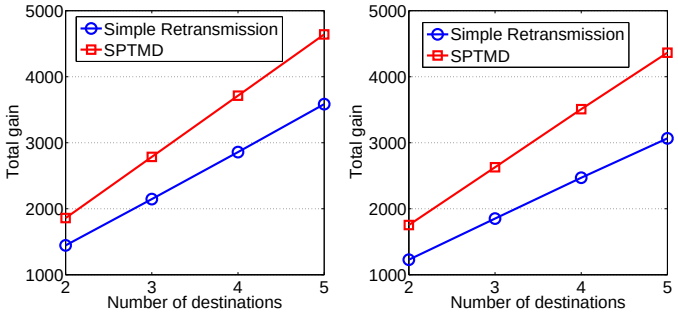


Fig. 11. Comparison between the gain of the simple retransmission and SPT mechanisms in the case of single packet transmission, $m = 10$, $k = 1$, $t = 10$; (a) $p \in [0.2, 0.4]$, (b) $p \in [0.2, 0.6]$.

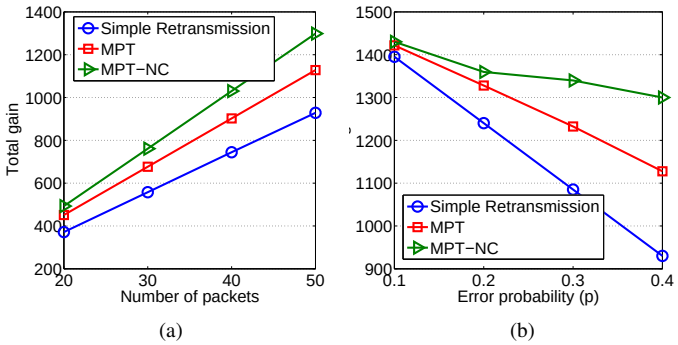


Fig. 12. Comparison between the gain of simple retransmission, MPT, and MPT-NC mechanisms, $m = 5$, $n = 1$; (a) $p = 0.4$, $t = 5$ (b) $k = 50$.

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 MPT-NC mechanism is more than that of the other methods. Moreover, the gain of the MPT mechanism is more than that of the simple retransmission method. Fig. 12(a) shows that the gain of the MPT-NC mechanism is up to 15% and 45% more than that of the MPT and simple retransmission methods, respectively. Also, the efficiency of the NC increases as we increase the number of packets which are coded together.

We evaluate the effect of link's error rate on the gain in Fig. 12(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 MPT and simple retransmission mechanisms drops dramatically as we increase the error rate. In contrast with the other methods, MPT-NC is more robust to the error rate, which is due to using NC.

We repeat the experiment of Fig. 12(a) in Fig. 13(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 Fig. 12(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

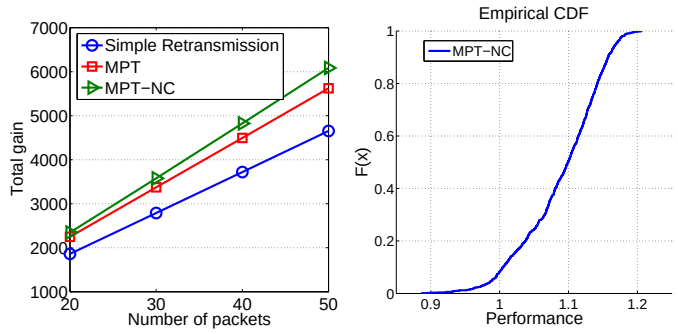


Fig. 13. Comparison between the gain of simple retransmission, MPT, and MPT-NC mechanisms, $m = 5$, $n = 5$, $p \in [0.3, 0.5]$; (a) total gain, $t = 5$ (b) Performance of the MPT-NC mechanism over the MPT method.

of packets. By comparing Fig. 12(a) with Fig. 13(a) we find that the difference between the MPT and MPT-NC decreases in the case of multiple destinations, which is because of the diversity of the links. Consequently, the efficiency of MPT-NC increases in the case that the error rates of the links are close to each other.

We compare the performance of the MPT-NC mechanism over the MPT in Fig. 13(b). For this purpose, we divide the gain of the MPT-NC mechanism by that of the MPT 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 MPT-NC mechanism is less than that of the MPT method. Moreover, in more than 50% of the cases, the number of delivered symbols of the MPT-NC protocol is more than 10% higher than that of the MPT mechanism.

VIII. 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 have a different weight. We first address the case of single packet transmission to a single destination node, and we show that the optimal solution of this problem has a round-robin pattern. Then, we extend our solution to the case of multiple 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-packets random linear NC. 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 mechanism. Moreover, using random linear NC can enhance the gain.

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] B. Zhao and M. Valenti, "The throughput of hybrid-ARQ protocols for the gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1971–1988, 2001.

[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*, vol. 52, no. 6, pp. 2551–2567, 2006.

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

[6] 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.

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

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

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XOs in the air: practical wireless network coding," in *ACM SIGCOMM*, 2006.

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

[11] ——, "Deadline-aware broadcasting in wireless networks with local network coding," in *IEEE ICNC*, Jan 2012.

[12] P. Ostovari, A. Khreishahand, and J. Wu, "Deadline-aware broadcasting in wireless networks with network coding," in *IEEE GLOBECOM*, Dec 2012.

[13] A. Khreishah, I. Khalil, P. Ostovari, and J. Wu, "Flow-based xor network coding for lossy wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2321–2329, 2012.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] P. Ostovari, J. Wu, and A. Khreishah, "Trade-off between redundancy and feedbacks in wireless network communication," *Ad-Hoc & Sensor Wireless Networks*, 2013.

[20] 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.

[21] 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.

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

APPENDIX

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

$$u = \sum_{i=1}^{m} w_i \times (1 - p^{x_i}), s.t \sum_{i=1}^{m} x_i = t$$

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

$$u = w_1 \times (1 - p^{x_1}) + w_2(1 - p^{x_2})$$
$$x_1 + x_2 = t$$

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 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 $p^{x_1} < \frac{w_1}{w_2} p^{x_2+1}$. Then, $p^{x_1-1} < \frac{w_1}{w_2} p^{x_2}$, and we have:

$$w_1 p^{x_1-1} < w_2 p^{x_2}$$
$$w_1(1-p)p^{x_1-1} < w_2(1-p)p^{x_2} \Rightarrow \Delta_{x_1-1} < \Delta_{x_2}$$

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: 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:

$$w_1 p^{x_1} > w_2 p^{x_2-1}$$
$$w_1(1-p)p^{x_1} > w_2(1-p)p^{x_2-1} \Rightarrow \Delta_{x_1} > \Delta_{x_2-1}$$

Therefore, it should be more efficient to increment x_2 in the previous state. Thus, in the current state we will have $(x_1 + 1, x_2 - 1)$, which for $x_2 \geq 1$ ($x_2 - 1$ cannot be negative) contradicts with 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 more gain. Lemma 2 is the reverse of Lemma 1, which results in a round-robin incrementing pattern. ■

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 SPT 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 $x_j, j \neq i$. ■