Priority-Based Broadcasting of Sensitive Data in Error-Prone Wireless Networks

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

***Abstract*—Providing reliable transmission in wireless communication networks is an important problem which is typically addressed using feedback and acknowledgment messages. In the networks where using feedbacks is not possible, such as real-time systems, an alternative approach is to maximize the possible gain that the destination nodes are expected to receive. In this paper, we consider transmission of data with different priorities, and study the problem of maximizing the total gain in the case that partial data retrieval is acceptable. We propose an optimal solution that benefits from network coding. We also consider the case of burst errors and discuss how can we make our proposed method robust to this type of error. We evaluate our proposed priority-based data transmission method using both simulations and results from the implementation on a USRP testbed.**

***Keywords*—Symbol-level coding, broadcasting, reliability, burst error, random linear network coding, priority, wireless networks, USRP testbed.**

I. INTRODUCTION

Broadcasting schemes are widely used for disseminating data and control messages in wireless networks. However, the error-prone wireless links in wireless networks create a challenge for reliable transmission. To handle this challenge, 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 (least significant bit). Therefore, any mechanism that addresses numeric data transmissions in a lossy environment should consider the weights of the bits.

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 and 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.

Consider a 2-digit decimal number in its Binary-Coded Decimal (BCD) representation. In BCD, each decimal digit is represented as a 4 bits binary number. For example, the BCD representation of 94 is 10010100, in which the four leftmost bits and the 4 rightmost bits represent 9 and 4, respectively. Assume that the error rate of the link between the source and a destination node is equal to 0.2. Moreover, we define each 4 bits as a symbol, and the source node can totally transmit 4 symbols. The gain of the user from each symbol is equal to the multiplication of receiving probability of the symbol by its importance. In this case, the weight of the symbols 9 and 4 are 10 and 1, respectively. As a result, the utility gain of the destination is equal to $10 \times (1-p^{x_1}) + 1 \times (1-p^{x_2})$. Here, x_1 and x_2 are the number of transmissions assigned to the most significant and least significant symbols, respectively, and p is the link's error rate. Table I depicts the possible distribution of the 4 transmissions to the 2 symbols, and the expected total gain of the destination node in each case. The table shows that, in our example, the optimal solution is to assign 3 and 1 transmissions to the symbols 9 and 4, respectively.

TABLE I. MOTIVATION EXAMPLE.

x_1	4	3	2	1	0
x_2	0	1	2	3	4
u	9.984	10.72	10.56	8.992	0.9984

In this paper, we find the optimal scheme to assign the transmission to the symbols with unequal priorities. Our contributions are:

- We study the problem of maximizing the total gain in the case of partial data delivery with unequal priorities.
- We propose the optimal solution to maximize the total gain, and we benefit from network coding in our solution. We also discuss how can we make our approach robust against burst errors.
- In addition to simulations, we report our results from implementing our method on a USRP testbed.

The remaining sections are organized as follows. We review the related work and describe linear network coding in Section II. The problem definition and the setting are provided in Section III. We propose our priority-based data transmission in Section IV. We evaluate the proposed mechanisms through simulations and real experiments in Section V. Finally, we conclude the paper in Section VI.

II. RELATED WORK AND BACKGROUND

A. Reliable Transmission

Feedback messages are commonly used in the reliable transmission methods over error-prone wireless communication networks. One of the most common approaches to providing a reliable transmission is Automatic Repeat reQuest (ARQ [1]. In order to reduce the transmission overhead of the ARQ method, hybrid-ARQ approaches [2], [5] are proposed, which combine ARQ with FEC (Forward Error Correction). Rateless (fountain) codes [3], [4] can be used to provide reliability without feedback messages. In these methods, the source generates and transmits an unlimited number of encoded packets until the destination nodes receive a sufficient number of encoded packets to be able to decode the coded packets and retrieve the original packets. In rateless codes, if the source has k original packets to send, a destination node needs to receive $N = (1+\epsilon)k$ [3] coded packets to decode the coded packets. Here, ϵ is a small number, which shows the overhead of the rateless codes. It is shown that this overhead goes to zero [6] as k goes to ∞. Because of their overhead for a small number of packets, rateless codes are not suitable for our problem, which is delay-sensitive and needs small batches of packets.

B. Network Coding

The authors in [7] introduce *Network Coding* (NC) [8]–[11] for wired networks. They show that NC achieves the capacity for the single multicast session problem. In [12], *Random linear network coding* is proposed, and the authors show that by selecting the coefficients of the coded packets randomly, we can achieve the capacity asymptotically with respect to the finite field size.

In random linear NC, coded packets are the linear combination of the original packets over a finite field, and the coefficients of the linear combinations are selected randomly. Any coded packet has a form of $\sum_{i=1}^{k} \alpha_i \times P_i$. Here, P and α are the packets and random coefficients, respectively. Packet P can be an original packet or a coded packet. Assuming that k packets are coded together, with a very high probability, any set of k random coded packets can be used to decode the coded packets and retrieve the original packets. This decoding process is done using the classic methods to solve a system of linear equations, such as Gaussian elimination.

NC is an effective method used to address reliable transmissions. Using NC, the source can transmit coded packets until it receives an acknowledgment from the destinations once they receive a sufficient number of coded packets to decode the coded packets. NC can also reduce the number of required transmissions to provide a reliable transmission. For example, the problem of one-hop reliable broadcasting work is studied in [13], [14]. In these works, the source node receives feedback from each destination node, showing their received packet. The source node benefits from NC, and combines the missed packets by the destination nodes in a way that each destination node can receive a missed packet using this coded packet. For instance, assume that packets P_1 and P_2 are transmitted by the source node, and destination nodes d_1 and d_2 receive packets P_1 and P_2, respectively, and miss the other packets. If we do not use NC, the source node needs to transmit both of the packets. However, the source node can transmit a linear combination of the packets instead.

The authors in [15] propose *symbol-level network coding* and show that it can increase the throughput compared to the packet-level network coding. In the case of symbol-level coding, even receiving a partial portion of the packets contributes to the utility, which increases the throughput. Later, the authors in [16], [17] benefit from the idea of symbol-level coding for distributing data and multimedia in vehicular networks.

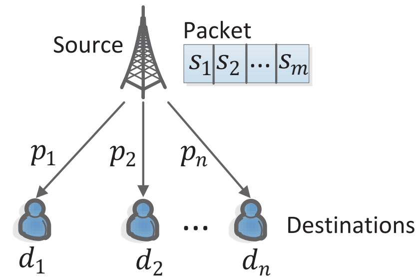


Fig. 1. System setting.

TABLE II. 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 (in the terms of the number of symbols)
$s_{j,i}$	The i-th symbol of the j-th packet
S_i	The i-th coded symbol
$x_{j,i}$	The number of transmissions of the coded symbols S_i
y_i	The number of transmissions of the coded i-th symbols S_i
P_i	The i-th packet
u_i	The gain (utility) from the i-th symbols in the case of non-coded symbols
u_i^{NC}	The gain (utility) from the i-th symbols when we use linear NC
u	The total gain (utility) in the case of non-coding
u^{NC}	The total gain (utility) of using linear NC

III. System Model and Problem Definition

A. System Setting

In our model, we have a single-hop wireless network, in which a source node broadcasts a batch of k packets to n destination nodes d. We assume that m symbols form a packet, and the symbols themselves might contain several bits. The priority of a symbol s_i is defined as the inverse of its weight w_i, and in general $w_i > w_{i+1}, \forall i: 1 \leq i \leq m-1$. The system model is shown in Figure 1. The set of symbols used in this paper is summarized in Table II.

We assume that each batch of packets has a deadline to be transmitted, and after that the source has another batch of packets ready to be transmitted. Consequently, we cannot use channel coding and hierarchical coding methods in our setting. The time assigned for transmitting a single packet is sufficient for t symbol transmissions. As a result, for a batch of k packets, the source node can transmit $t \times k$ symbols. If we do not consider a deadline for the packets, or assume that the source has infinite packets to transmit, the optimal solution can be found by way of a simple extension of the well-known channel coding theory [18].

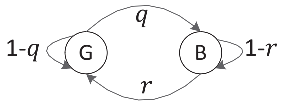


Fig. 2. Gilbert-Elliott model: modeling transmission errors using a 2-state Markov chain.

Our goal in this work is to maximize the total weight of the received symbols of a batch of k packets by the destination nodes. We use boolean variable $z_{j,i,l}$ to represent success and failure in delivering symbol s_i of the k-th packet to destination d_l. The value of the variable is equal to 1 in the case of successful delivery; otherwise the value is 0. We can model our objective function as the following utility function:

$$u = \sum_{j=1}^{k} \sum_{i=1}^{m} \sum_{l=1}^{n} z_{j,i,l} \times w_i$$

B. Error Model

We assume that the transmission errors are bursty. A burst error is a contiguous sequence of errors. In order to model the burst errors we use the Gilbert-Elliott model [19], [20]. The Gilbert-Elliott model is widely used for describing burst error patterns in data transmission channels. This model is based on a Markov chain with two states *good* and *bad*, represented as G and B, respectively. As shown in Figure 2, the system transits from state G to B with a probability equal to q. The probability of the transition from state B to G is equal to r. In the general form of the model, there are different error probabilities for the 2 states. However, in the simplified model, the error probability of states G and B are supposed to be 0 and 1, respectively. In the simplified version, the error probability is equal to the stationary state probability of being at state B, which is equal to $p = q/(q + r)$.

Figures 3(a)-(d) show the simulation results of 200 transmissions using the simplified version of the Gilbert-Elliott model. In Figure 3(a), q and r are set to 0.1. As a result, the average length of the burst errors and successful transmissions are the same. Figures 3(b) and (c) show that the average length of the burst errors and successful transmissions decrease as we increase q and r, which is due to the increase in the transition probability. In Figure 3(d), q and r are equal to 0.1 and 0.5, respectively. As a result, the lengths of consecutive successful transmissions are more than those of the burst errors.

IV. Priority-Based Data Transmission

Without network coding, the probability of symbol $s_{j,i}$ to be received by the l-th destination is equal to $1 - {p_l}^{x_{j,i}}$, where $x_{j,i}$ is the number of transmissions of symbol $s_{j,i}$. Also, the weight of the i-th symbol of the packets are equal. As a result, in the case of non-coding, the total gain (utility) becomes:

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

The problem of this scheme is that a destination node might receive some of the symbols multiple times, and might not receive the other symbols. This problem is known as the coupon collector problem. In order to solve this problem, we can benefit from random linear NC. We can code the i-th symbols of the k packets together as shown in Figure 4. Using this scheme, a destination node will be able to decode the i-th coded symbols and retrieve the original symbols once it receives at least k coded symbols. Thus, each coded symbol contributes the same amount of information to the destination nodes. 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. If we represent the number of transmissions assigned to the coded symbols over the i-th symbols of the packets as y_i, the utility becomes:

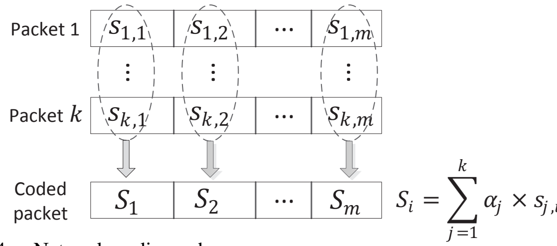


Fig. 4. Network coding scheme.

$$u^{NC} = \sum_{i=1}^{m} w_i \times k \sum_{l=1}^{n} \Big[\sum_{j=k}^{y_i \times k} \binom{k \times y_i}{j} \times (1 - p_l)^j \times p_l^{y_i \times k - j} \Big]$$

The right most summation calculates the decoding probability of the i-th symbols. A destination node can decode the i-th code symbols if it receives at least k coded symbols. Random linear NC decreases the probability of receiving partial symbols. This is because if a destination node receives less than k coded symbols, it will not be able to decode the coded symbols. As a result, depending on the assigned redundancy to each set of symbols of the k packets, using random linear NC might be efficient or not.

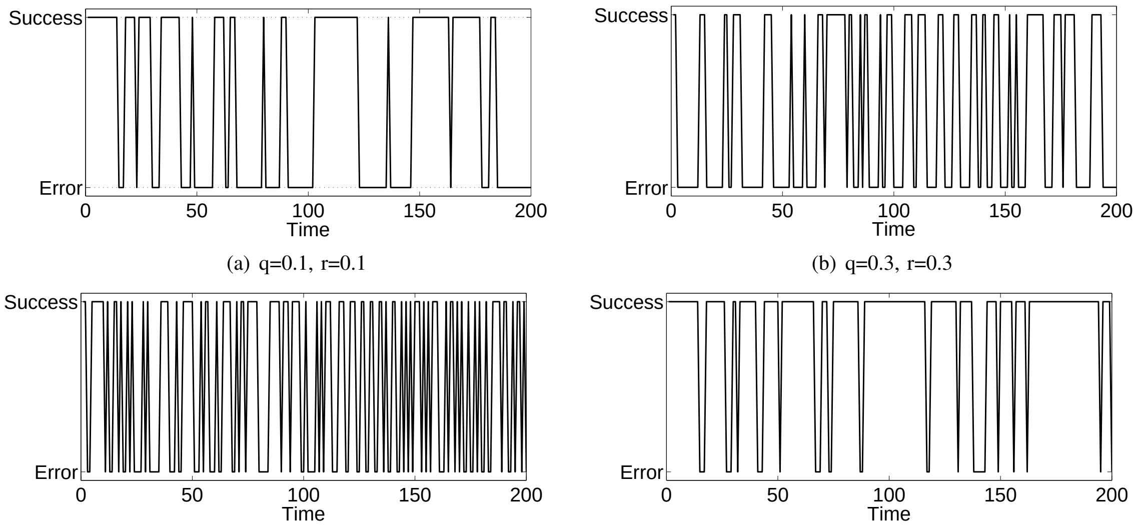
Consider a source node that wants to transmit two packets with a single symbol to a destination node, and the error rate of the link between these nodes is equal to 0.4. Assume that the number of transmissions is equal to 4. In the case of non-coding transmissions, the probability of receiving both of the symbols is equal to $(1-0.4^2)\times(1-0.4^2) = 0.7056$. Also, the probability of receiving just one of them is equal to $2 \times (1 - 0.4^2) \times 0.4^2 = 0.2688$. Therefore, assuming that the weight of the symbols is equal to 1, the total gain is $0.7056 + 0.2688 = 0.9744$. Using NC, these probabilities become $1 - 0.4^4 - 4 \times 0.4^3 = 0.7184$ and 0 respectively. Thus, using network coding in this example in not beneficial.

Following the above discussion, it is clear that none of the coding and non-coding schemes give us the optimal solution. Thus, we need a method that switches between them to finds the optimal scheme. The optimization in this case becomes:

$$\max \sum_{i=1}^{m} max(u_i, u_i^{NC})$$

$$s.t. \quad \sum_{j=1}^{k} \sum_{i=1}^{m} x_{j,i} = t$$

$$\sum_{i=1}^{m} y_i = t$$



(a) q=0.1, r=0.1 (b) q=0.3, r=0.3 (c) q=0.5, r=0.5 (d) q=0.1, r=0.5

Fig. 3. Simulation result for 200 transmissions using the Gilbert-Elliott model.

where:

$$u_i = \sum_{j=1}^{k} \sum_{l=1}^{n} w_i \times (1 - {p_l}^{x_{j,i}}) \tag{1}$$

$$u_i^{NC} = w_i \times k \sum_{l=1}^{n} \Big[\sum_{j=k}^{y_i \times k} \binom{k \times y_i}{j} (1 - p_l)^j \times p_l^{y_i \times k - j} \Big] \tag{2}$$

A straightforward solution is to apply brute search. For this purpose, we can find the optimal solution by calculating the gain in the case of coding and non-coding for all of the possible transmission distributions, and by selecting the distribution that results to the maximum gain. However, it is clear that the time complexity of the brute search is huge. In order to reduce the time complexity, we can benefit from the following lemma:

Lemma 1: For the distribution that results in the optimal solution we have $x_i \geq s_{i+1}, \forall 1 \leq i \leq m-1$

Proof: Both of the functions 1 and 2 are non decreasing functions. Therefore, assigning more transmissions to the symbols with a greater w_i results in a higher utility. ■

Lemma 1 helps us to reduce the search space dramatically.

Theorem 1: The number of ways that we can distribute t transmissions among m symbols using lemma 1 is in the order of t^m.

Proof: The number of ways that we can distribute t transmissions among m symbols using lemma 1 is equal to the number of ways to partition the integer number t to at most m partitions. The latter problem is referred to as the restricted version of the *integer partitioning* problem. In [21], it is shown that the number of distinct possible partitionings is in the order of t^m. ■

Theorem 1 shows that the time complexity of checking all the possible transmission distributions that see lemma 1 is exponential. However, in our problem, m is fixed. Moreover, a typical number of symbols is usually a small number, e.g. 4 or 5. As a result, the number of possible distributions that need to be checked is polynomial in terms of t.

Based on the discussion, we propose the Priority-Based Transmission (PBT) algorithm as follows. The PBT algorithm checks all the possible distributions that see lemma 1. For each possible distribution, we check the gain of the i-th symbols of the k packets in the case of coding and non-coding symbols, and calculate the maximum total possible gain for all of the symbols. We repeat this process for all of the possible transmission distributions and select the distribution that results in the maximum total utility. The details of the PBT method are shown in Algorithm 1.

After finding the optimal number of transmissions for each symbol, the source transmits each symbol several times. As mentioned in Section III, the errors in our model have a burst pattern. Consequently, putting the transmissions of the same symbol beside each other is not logical. The reason is that, if an error happens in a transmission, there is a large chance that the next transmission faces an error as well. On the other hand, a successful transmission might be followed by another successful transmission with a high probability. Consequently, if the source node transmits a symbol multiple times, each following the other, it is with a high probability that a destination node might receive the symbol multiple times or might miss all of the transmissions assigned to that symbol. In the former case, the transmissions are wasted, as some of these successful transmissions could be assigned to other symbols. In the later case, the destination node will not receive and gain from that particular symbol.

In order to make the PBT method robust against burst errors, we transmit the different symbols in a round-robin patter. In PBT, the source node starts from the most important symbol of the packets, transmits each symbol of a packet once, and subtracts one transmission from the number of transmissions assigned to each symbol. It then repeats the process until no more transmissions are left. Our results from the implementation on the USRP testbed confirm the effectiveness of the round-robin transmission patter.

Algorithm 1 PBT Algorithm

Initialize: $maxGain = 0$, $argmax = 1$
for $i = 1 : m$ **do**
  $coding(i) = 0$
for each distribution **do**
  for i=1:m **do**
    calculate u_i^{NC} and u_i using 2 and 1
    if $u_i^{NC} \geq u_i$ **then**
      $gain = gain + u_i^{NC}$
    else
      $gain = gain + u_i$
  if $gain \geq maxGain$ **then**
    $maxGain = gain$
    $argmax = index$
    for i=1:m **do**
      coding(i)=1

V. Experiments and Evaluation

In the following sections, we first report our simulation results. We then present our real experiment results.

A. Simulations

1) Setting: In order to evaluate our method, we implemented a simulator in the MATLAB environment. We compare our proposed PBT (priority-based transmission) method with a *simple retransmission* (SR) method and MPT-NC method [22]. In the SR method, we distribute the transmissions evenly to the different symbols. The MPT-NC method, finds the optimal distribution of the transmissions in the case of non-coding. It then checks whether applying network coding among the symbols that have the same weight can increase the gain or not. In the former case, the MPT-NC method enables coding for these set of symbols. We run each simulation for 100 random topologies with different links' error rates. Also, we repeat the simulation of each random topology 100 times. In our simulations, the weight of the i-th symbol of each packet is equal to 2^{m-i}. In the simulations, the number of destinations, packets, and symbols of each packet are equal to 10 and 5, and 5, respectively.

2) Results: In the first experiment, we measure the effect that the number of transmissions has on the total gain in Figure 5(a). For each random topology, r is set to 0.12. Moreover, q for each link to the destinations is randomly set in the range of $[0.05, 0.12]$. As a result, the error rate of the links are in the range of $[0.2941, 0.5]$. As expected, the total gain in Figure 5(a) increases as we increase the number of transmissions. In this figure, the total gain of the PBT method is up to 16% and 50% more than that of the MPT-NC and SR methods, respectively.

In Figure 5(b), we reduce the average burst error size by half, and repeat the previous experiment. For this purpose, we multiply r and the range of q by 2. Therefore, the range of error rates is still $[0.2941, 0.5]$. Although the error rates in Figures 5(a) and (b) are the same, the total gain in Figure 5(b) is more than that of in Figure 5(a). The reason is that, the size of burst errors in Figure 5(b) is smaller that in Figure 5(a). However, the reduction in the burst size has more of an effect

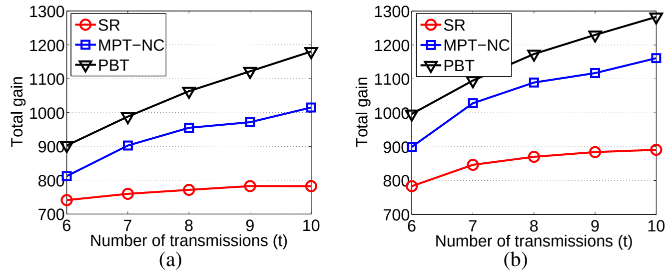


Fig. 5. Effect of number of transmissions on the total gain . $m = 5$, $k = 5$, $n = 10$. (a) $r = 0.12$, $q \in [0.05, 0.12]$. (b) $r = 0.24$, $q \in [1, 0.24]$.

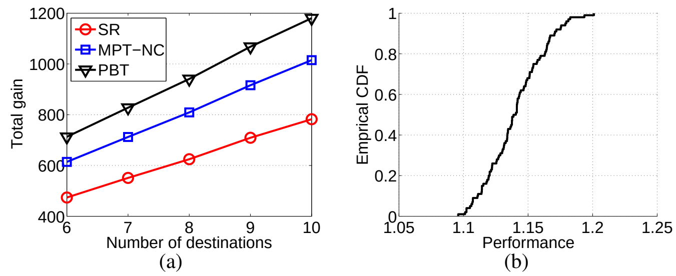


Fig. 6. $r = 0.12$, $q \in [0.05, 0.12]$, $m = 5$, $k = 5$, $n = 10$. (a) Total gain. (b) Empirical CDF of the performance, $n = 10$.

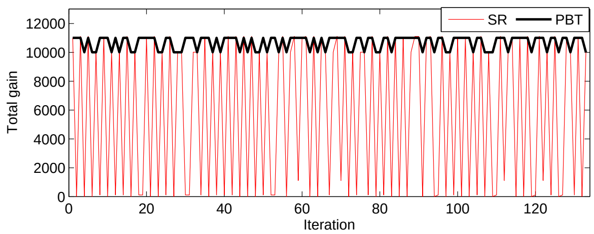
on the gain of MPT-NC and SR methods, as the PBT method is more robust against the burst errors.

In Figure 6(a), r is set to 0.12, and q is in the range of $[0.05, 0.12]$. Also, the number of symbol transmissions t is equal to 10. The figure shows that the total gain of the methods is almost linear with respect to the number of destination nodes. Figure 6(b) shows the empirical CDF of our method's performance. We define the performance as the division of the total gain of the PBT method by that of the MPT-NC scheme. The figure depicts that in 20% of the cases, the performance of our method is between 1.16 and 1.2. Moreover, in 50% of the runs, the performance of the PBT method is up to 1.14.

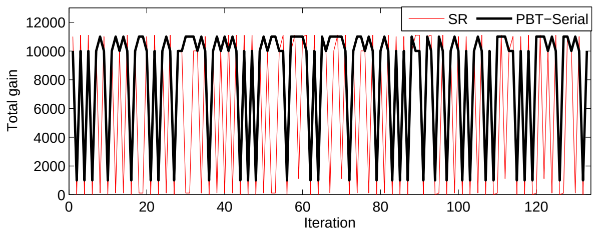
B. Real Experiment

For the real experiments we use 3 USRPs (Universal Software Radio Peripheral) to evaluate our proposed PBT method. One USRP is the sender, one is the receiver, and the other one works as an interfering node. The devices work on the narrowband, and the central frequency is 1.26GHz. The antenna gain on each node is 20 db. We send the packets for one minute and compare the received packets using the PBT and the SR approaches.

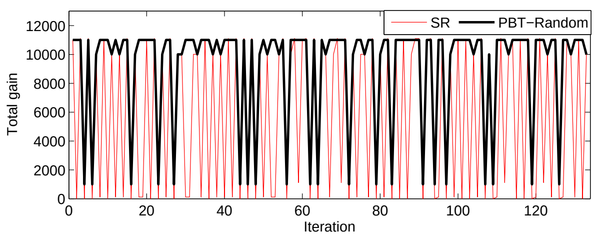
The source node transmits a 5-digit binary coded decimal (BCD) number, in which the weight of the i-th digit (from left to right) is equal to $10^{5-i}, \forall\ 1 \leq i \leq 5$. We repeat the same experiment several times (iterations). Figure 7(a) shows the total gain of the PBT and SR methods for 134 runs. The bold and thin lines show the total gain of the PBT and simple retransmission, respectively. In the PBT method, we put the symbols in a round-robin fashion. In other words, we first put the symbols to be sent beside each other, and then we put the redundant symbols. As depicted in Figure 7(a), the total gain of the PBT method is almost always more than or equal to that of the SR. The oscillation in the gains is because of the randomness of the channel, which results in errors happening in different parts of the data in different runs.



(a)



(b)



(c)

Fig. 7. Real experiment. Transmission of a 4-digit BCD number. $k = 1$, $t = 5$, $p = 0.2442$. (a) PBT method. (b) PBT with serial symbols pattern. (b) PBT with random symbols pattern.

In Figure 7(b), we modify the proposed PBT method by changing the round robin patter with a serial pattern. It means that, we put symbol s_1 for x_1 times beside each other. Then, we put the second symbol x_2 times and so on. As Figure 7(b) depicts, there are cases in which the gain of the SR method is more than that of the PBT method, which is due to the bursty errors. If a bursty error happens, all of the repeated serial symbols might be lost. On the other hand, if a symbol is received correctly, then the destination might receive that symbol several times, which is useless. It suggests that repeating the symbols in a serial patter is not an appropriate approach.

The total gain of the methods in the case of a random symbol pattern is shown in Figure 7(c). As the figure depicts, in some cases both of the methods results in a similar total gain. However, in many cases, the gain of the PBT method is more than that of the SR method. By comparing the Figures 7(a)-(c) we find that the gain of the PBT method with a random symbol patterns is in between that of the serial and round-robin pattern.

VI. Conclusion

Providing reliable transmission in wireless communication networks is critical. In this paper, we consider transmission of data with different levels of importance. Instead of ensuring the reception of all of the packets by the destination nodes, we are interested in maximizing the utility that the designation nodes will gain in the case of partial retrieval of data. We propose an optimal solution for assigning transmissions for different parts of the data to be transmitted, which benefits from network coding. In our priority-based transmission method, we consider the possible burst errors. We implemented our method on a USRP testbed. We evaluated our method both through simulations and the results from the real testbed.

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] 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.

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

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

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

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

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

[11] 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.

[12] 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.

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

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

[15] 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.

[16] 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.

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

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

[19] E. N. Gilbert, Capacity of a burst-noise channel, Bell system technical journal 39 (5) (1960) 1253–1265.

[20] E. Elliot, Estimates of error rates for codes on burst-nise channels, Bell system technical journal 42 (5) (1963.) 1977–1997.

[21] H. Gupta, Tables of partitions, Cambridge University Press, 1958.

[22] P. Ostovari, J. Wu, A. Khreishahand, Efficient symbol-level transmission in error-prone wireless networks with network coding, in: IEEE WoWMoM, 2013, pp. 1–9.