A Cooperative Scheme for Shaping Degree Distribution of LT-Coded Symbols in Network Coding Multicast

Apirath Limmanee and Werner Henkel
Jacobs University
EECS, TrSys, Campus Ring 1
28759 Bremen, Germany
Email: {a.limmanee, w.henkel}@jacobs-university.de

***Abstract*—This paper shows that LT-encoding at the source node and network coding at intermediate nodes cannot be applied together sequentially in binary erasure channels (BECs) without significant receiver performance degradation due to the distortion of the degree distribution in received LT-coded symbols. Two countermeasures are discussed. The first one is wise assignment of network codes, which totally eradicates the distortion in some specific, but not all, cases. The second one, being more universal, is a cooperation between the source and the relay nodes. The source introduces buffers for temporarily storing LT-coded output prior to transmission. Each buffer is a first-in-first-out (FIFO) queue associated with one outgoing edge of the source node. It is shown that under some conditions related to the degree of a particular LT-coded symbol, it is better to put that symbol into one buffer instead of another in order to keep the degree distribution distortion low. In addition, we paradoxically discover that the relay node, instead of always performing network coding, can improve the receiver performance by discarding some LT-coded symbols with a certain probability, since this will reduce the degree distribution distortion.**

I. Introduction to LT and Network Codes in Relay Network Multicast

LT codes or other rateless codes as a forward error correction (FEC) scheme are arguably most useful in multicast applications in which its one-to-many nature makes the acknowledgement scheme very unpleasant. LT codes can potentially generate an infinite stream of coded output symbols such that, even when some symbols are erased, the receiver can recover k original symbols using any K received coded symbols when K is only slightly larger than k.

Like LT codes, network codes find their first and simplest application in multicast. However, both types of codes have opposite aims. While LT codes increase redundancy in the networks to compensate for erasures, network codes decrease it by means of coding at the bottlenecks.

According to Fig. 1, where A aims to multicast two binary symbols b_1 and b_2 to D and E, we can see that the node F encodes b_1 and b_2 together to achieve the multicast rate of 2 bits per unit time, if each edge represents the capacity of one bit per unit time. D receives b_1 from the path ABD and can recover b_2 from the symbol $b_1 \oplus b_2$ from $ACFGD$, whereas E receives b_2 from the path ACE and recovers b_1 from the symbol $b_1 \oplus b_2$ from $ABFGD$. Had network coding not been there, only either b_1 or b_2 would have been able to pass the bottleneck FG in one unit time, i.e., one receiver would have been unable to use one of its possible transmission paths.

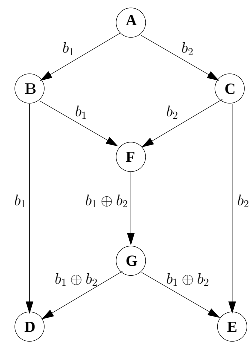

Fig. 1. Network coding in a butterfly network

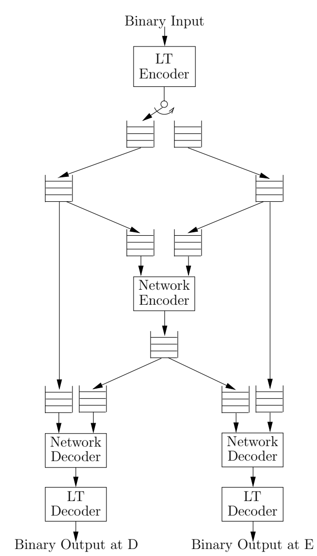

Fig. 2. Detailed system model including encoding and decoding blocks as well as the buffer structure

In this paper, we assume that the source node generates

ISBN 978-3-8007-3211-1

LT-coded symbols which are subsequently network coded by intermediate nodes along their ways to the receivers. To make things more concrete, Fig. 2 displays a block diagram showing all encoding and decoding processes as well as the buffer structure from the source to the destinations in accordance with the network in Fig. 1.

One can observe a switch placed after the LT-Encoder block and prior to two buffers. While traditional network coding only requires that the switch turns to each buffer half of the time, it is not the case in this paper, in which the switching decision depends on the degree of the current LT-encoding output. This is one major discovery in this work.

Despite having our own system shown in Fig. 2, we are aware of previous works dealing with similar problems. Therefore, we present a short review of their works in the next section.

After that, Section III elaborates on important parts in our system, such as LT encoder and decoder. The most crucial function influencing the LT-decoder performance is the degree distribution which must be carefully chosen by the encoder. Section IV, however, shows that the well-designed degree distribution can easily be distorted by network coding in binary erasure channels (BECs), leading to a significant degradation in receiver performance.

Section V provides a solution to the problem for some special cases by means of wise assignment of network codes. In other cases, however, we need a cooperative scheme proposed in Section VI to improve the receiver performance, as shown in Section VII.

II. Literature Review

Although many recent works regarding network coding focus on two-way wireless relay networks rather than relay network multicasts [4,5,6], it is suggested in [5] that the problem of two-way relay networks, and of information exchange in general, can be transformed into a multicast problem via some graph transformations. Thus, the study of information multicast in this work might lead to further discoveries in more generalized cases.

In addition, main components used in those systems are identical or similar to ours. The decode-and-forward (DF) scheme used in [6] employs a channel encoder at each transmitter and a network encoder at the relay, which is structurally identical to our system. However, we use LT codes instead of turbo codes and therefore need no decoder at the relay. This is similar to distributed LT codes proposed in [3], but the receiver in that system receives information only from the relay, whereas our sink receives it via the direct path as well. Since LT codes are erasure correcting codes, binary erasure channels (BECs) are used in our model instead of Gaussian channels.

Like in our system, the buffering scheme is considered important and given careful attention in [5,7].

III. LT and Network Encoding and Decoding

Three steps are needed to generate an LT-coded symbol. Firstly, a degree d is selected from a degree distribution, which is a discrete probability density function mapping a degree to the probability that the degree is selected. Secondly, d original symbols are chosen uniformly at random. Finally, an output symbol is derived by XORing all chosen symbols from the previous step [2].

In this paper, we use a robust soliton distribution, which is constructed such that the failure probability of the message-passing decoder is δ for a given number $K = k + O(\sqrt{k} \cdot \ln^2(k/\delta))$ of received symbols [2].

Definition 1: The robust soliton distribution (RSD) $\mu(i)$ is derived from the normalization of two functions $\rho(i)$ and $\tau(i)$ as

$$\mu(i) = \frac{\rho(i) + \tau(i)}{\beta}, \ 1 \leq i \leq k, \tag{1}$$

where

$$\rho(i) = \begin{cases} 1/k, & i = 1 \\ 1/(i(i-1)), & 2 \leq i \leq k, \end{cases} \tag{2}$$

$$\tau(i) = \begin{cases} R/(i \cdot k), & 1 \leq i \leq (\text{round}(k/R)) - 1, \\ (R\ln(R/\delta))/k, & i = k/R, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

$$R = c \cdot \sqrt{k} \cdot \ln(k/\delta), \tag{4}$$

$$\beta = \sum_{i=1}^{k} (\rho(i) + \tau(i)). \tag{5}$$

The paramater c in (4) is a suitable non-negative constant used in the design, whereas δ is the failure probability mentioned earlier [2].

The robust soliton distribution with parameters $k = 1000$, $c = 0.1$, and $\delta = 0.5$ is shown in Fig. 3. The most important observation regarding the plot is that it has two peaks at 2 and 42.

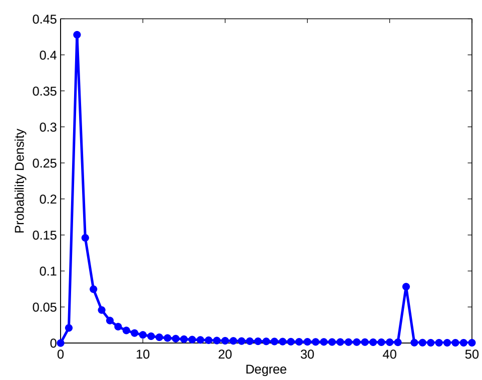

Fig. 3. Robust soliton distribution with $k = 1000$, $c = 0.1$, and $\delta = 0.5$

Now, let us consider the network in Fig. 1. Assume that node A would like to multicast LT-coded outputs b_1 and b_2 with the robust soliton distribution shown in Fig. 3 to D and E. After the node D receives b_1 and $b_1 + b_2$, it first deducts

b_1 from $b_1 + b_2$ to obtain b_2. We call this deduction "network decoding". After that, both b_1 and b_2 are LT-decoded by the message-passing algorithm, as described in [2], to obtain the original message.

Unfortunately, when network coding is used in severe erasure channels, the well-designed degree distribution can be distorted before LT-coded symbols reach the destination. The next section addresses this issue as well as the resulting receiver performance deterioration.

IV. Degree Distribution Distortion and Receiver Performance Deterioration in Binary Erasure Channels

Suppose that the edge BD is a binary erasure channel having an erasure probability of 0.1 whereas other edges are erasure-free. Thus, on average, once every ten times the receiver D does not have b_1 to deduct from b_1+b_2 to complete the network decoding process. Instead, $b_1 + b_2$ enters directly into the LT decoding process. Therefore, on average, for every ten symbols of b_1 and ten of b_2 transmitted, D receives nine b_1, nine b_2 (after network decoding), and one $b_1 + b_2$. If b_1 and b_2 follow the robust soliton distribution $\mu(i)$, the overall degree distribution becomes

$$\psi(i) = \frac{18}{19}\mu(i) + \frac{1}{19}\varphi(i), \tag{6}$$

where $\varphi(i)$ is the degree distribution of $b_1 + b_2$.

It is precisely the $\varphi(i)$ that alters the overall degree distribution of D's received symbols from the robust soliton case. Since, in practice, the number k of LT input symbols is large, we can assume that the LT-encoded symbols b_1 and b_2 are not made up of some common original symbols. This allows us to approximate the overall degree distribution at D after network decoding as

$$\psi(i) \approx \begin{cases} \frac{18}{19}\mu(i), \ i = 1 \\ \frac{18}{19}\mu(i) + \frac{1}{19}\sum_{j=1}^{i-1}\mu(j)\mu(i-j), \ \text{otherwise.} \end{cases} \tag{7}$$

Figures 4 and 5 compare the plot of the robust soliton distribution $\mu(i)$ in the erasure-free case with the distorted distribution due to erasures at BD obtained by the analytical approximation $\psi(i)$ in (7) and that obtained from simulation by counting the degree of 10^7 symbols received at D around the two peaks at 2 and 42, respectively. We can see that the steepness of the peaks are lessened in the distorted case. Moreover, an unwanted peak is formed at 44.

Although the distortion of the degree distribution is small, its effect on the performance is clearly visible. Fig. 6 shows the histogram of the number of LT-coded symbols needed to be transmitted until the original symbols can be recovered by D. Fig. 7 makes a comparison between the number of symbols needed to be transmitted when the erasure probability of BD is 0.1 in the network in Fig. 1 and that when the erasure probability is 0.05 in normal point-to-point communication. Although the average erasure probabilities in both cases are the same, the network coding case requires more symbols due to the distortion of the degree distribution.

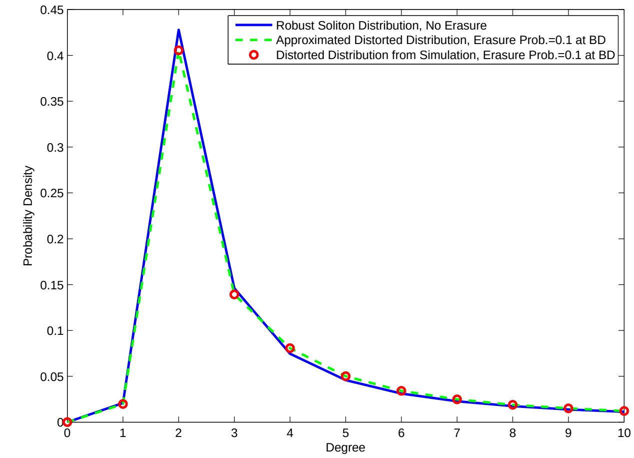

Fig. 4. Robust soliton distribution around the first peak in comparison with distorted distribution caused by erasures at the edge BD

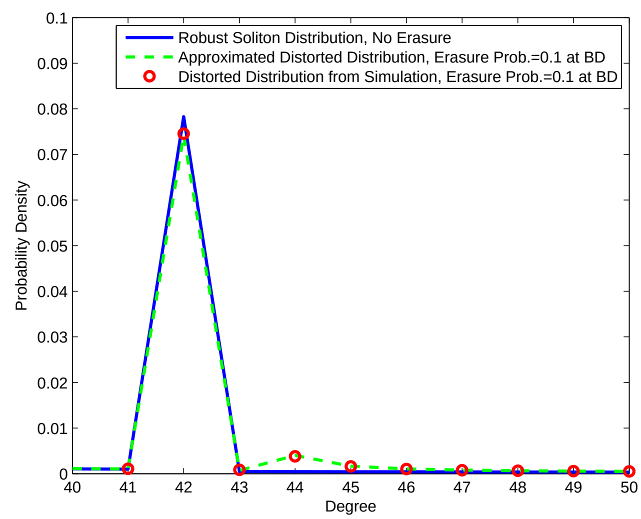

Fig. 5. Robust soliton distribution around the second peak in comparison with distorted distribution caused by erasures at the edge BD

V. The First Solution: Wise Assignment of Network Codes

In case erasures occur only along the edge BD with the erasure probability of 0.1, the solution to the degree distribution distortion problem is very simple. We change our network codes from those in Fig. 1 into those in Figure 8. Since we now transmit $b_1 + b_2$ instead of b_1 along BD, when $b_1 + b_2$ is erased, D only receives b_1, yielding no distortion, when $b_1 + b_2$ is not erased, D can network-decode and receive both b_1 and b_2, yielding no distortion either.

This solution can be performed by the source alone. When the source A is informed by B that there are severe erasures at BD, it simply transmits $b_1 + b_2$ instead of b_1 to B while other nodes just work as usual. In general, this solution is applicable

ISBN 978-3-8007-3211-1

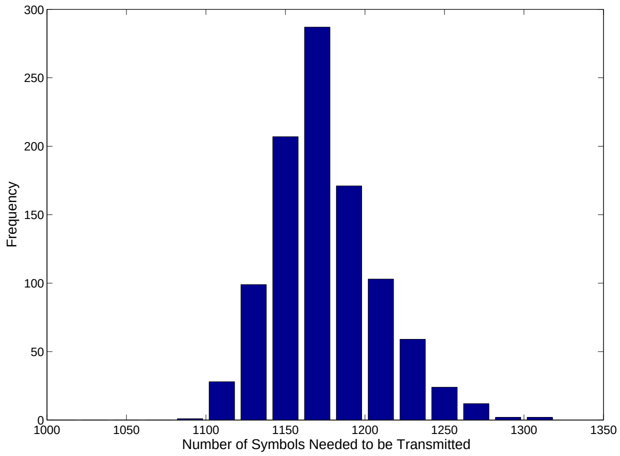

Fig. 6. Histogram of the number of LT-encoded symbols needed to be transmitted such that all original symbols are recovered

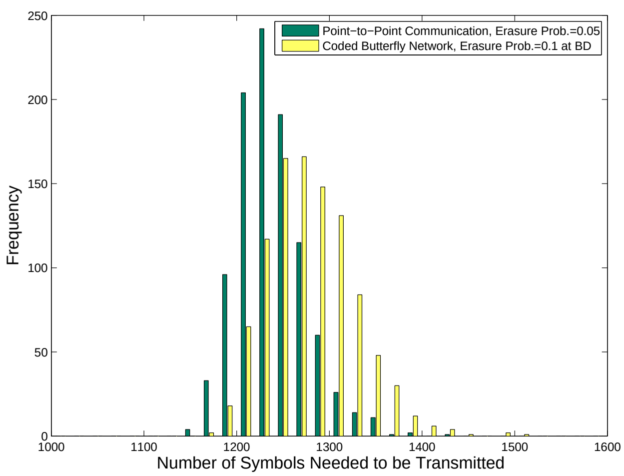

Fig. 7. Comparison of histograms of the number of LT-encoded symbols needed to be transmitted such that all original symbols are recovered in two cases, point-to-point communication with the erasure probability of 0.05 and coded butterfly network with the erasure probability of 0.1 at BD

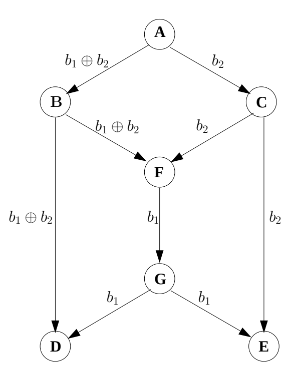

Fig. 8. Network coding in a butterfly network when only BD is erasure-prone

when erasures occur along a single edge, with the transmission pattern depending on which edge is erasure-prone, as shown in Table 1.

TABLE I
RECOMMENDED TRANSMISSION WHEN AN EDGE IS ERASURE-PRONE

Erasure-prone Edge	Choices of Recommended Transmission to (AB, AC)
AB or BF or GE	(b_1, b_2), (b_2, b_1), $(b_1 + b_2, b_1)$, $(b_1 + b_2, b_2)$
AC or CF or GD	(b_1, b_2), (b_2, b_1), $(b_1, b_1 + b_2)$, $(b_2, b_1 + b_2)$
BD	$(b_1 + b_2, b_1)$, $(b_1 + b_2, b_2)$
CE	$(b_1, b_1 + b_2)$, $(b_2, b_1 + b_2)$
FG	(b_1, b_2), (b_2, b_1)

The next section deals with a more general case in which more than one edge is erasure-prone.

VI. THE COOPERATIVE SOLUTION PERFORMED BY THE SOURCE AND THE RELAY

As an example, consider the case when both the edges BD and CE are erasure-prone. In this case, we cannot find a solution from the previous section that satisfies both receivers. If A chooses to transmit (b_1+b_2, b_2) to (AB, AC), the receiver D is satisfied but E still suffers from the degree distribution distortion.

It can be imagined that the ideal solution can be found such that, instead of transmitting b_1 and b_2 with the robust soliton distribution, we use another degree distribution $\nu(i)$ which, after being distorted by erasures, becomes the robust soliton distribution. However, such a distribution is very difficult, if not impossible, to be found. Indeed, it is proven in a similar work [3] that one cannot find a degree distribution $\nu(i)$ for b_1 and b_2 such that b_1+b_2 follows the robust soliton distribution.

Since we cannot achieve this ideal solution, we will offer a cooperative scheme performed by the source node A and the relay node F that corrects the degree distribution distortion only in the positions that most affect the performance. Those positions are at degrees 2, 4, 44, and 84.

The severe distortion at 4, 44, and 84 is due to the high probability that b_1 and b_2 have the degrees of 2 or 42, causing b_1+b_2 to have the degree of 4, 44, or 84 with higher probability than what is required, whereas the distortion at 2 is due to the fact that $b_1 + b_2$ cannot have the degree 2 unless both b_1 and b_2 have the degree of 1, which is unlikely.

According to our cooperative scheme, the distortion at 44 and 84 is reduced by applying a buffering scheme at the source. The scheme arranges the LT-encoder output in such an order that b_1 and b_2 with degrees of 2 and 42, 42 and 2, or 42 and 42 are not allowed to be simultaneously transmitted and mixed at the relay thereafter, as implemented in Step 4) of Algorithm 1. In addition, the distortion at 2 and 4 is corrected by selectively discarding some symbols at the relay, i.e., when both b_1 and b_2 have the degree of 2, there is a probability p_c that the relay performs network coding and a probability $1 - p_c$ that either b_1 or b_2 is transmitted while the other is discarded. This will relieve the excess of symbols

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach

with degree 4 and the shortage of those with degree 2. The scheme is implemented in Algorithm 2.

Algorithm 1: The buffering scheme performed by the source

Let

- $S_l = \begin{cases} 1 \text{ if the switch is pushed to the left buffer} \\ 0 \text{ if the switch is pushed to the right buffer} \end{cases}$,
- λ_n be the LT-encoder output at the discrete time $n = 0, 1, 2, ..., N_{max}$,
- $d(\lambda_n)$ be the degree of λ_n,
- N_B be the size of each buffer,
- $B_l(m), m = 0, 1, 2, ..., N_B - 1$ be the m^{th} element in the left buffer such that $B_l(m)$ precedes $B_l(m + 1)$,
- $B_r(q), q = 0, 1, 2, ..., N_B - 1$ be the q^{th} element in the right buffer such that $B_r(q)$ precedes $B_r(q + 1)$,
- π_l, π_r be the pointers of the left and the right buffer, respectively,
- and b_1, b_2 be the current symbol to be transmitted to the edges AB and AC, respectively,

1) Initialize
 $n := 0$
 $S_l := 1$
 $B_l(0) := \lambda_0$
 $\pi_l := 1$
 $\pi_r := 0$
 $\{B_l(m)|m = 1, 2, ..., N_B - 1\} := \emptyset$
 $\{B_r(q)|q = 0, 1, 2, ..., N_B - 1\} := \emptyset$
2) $n := n + 1$
3) if $S_l = 0$ {
  $S_l := 1$
  if $\pi_l < N_B$ {
   $B_l(\pi_l) := \lambda_n$
   $\pi_l := \pi_l + 1$ }}
4) while $S_l = 1$ {
  if $(d(\lambda_n), d(B_l(\pi_r))) \notin \{(2, 42), (42, 2), (42, 42)\}$ {
   $S_l := 0$
   if $\pi_r < N_B$ {
    $B_r(\pi_r) := \lambda_n$
    $\pi_r := \pi_r + 1$ }}
  else {
   if $\pi_l < N_B$ {
    $B_l(\pi_l) := \lambda_n$
    $\pi_l := \pi_l + 1$ }}}
5) if the channel access is allowed {
  $b_1 := B_l(0)$
  $b_2 := B_r(0)$
  $B_l(m) := B_l(m + 1),\ m = 0, 1, 2, ..., N_B - 1$
  $B_r(q) := B_r(q + 1),\ q = 0, 1, 2, ..., N_B - 1$
  $\pi_l := \pi_l - 1$
  $\pi_r := \pi_r - 1$
  Transmit b_1 and b_2. }
6) if $n < N_{max} - 1$ {
  Go back to 2). }
 else {
  Exit. }

Algorithm 2: The discarding scheme performed by the relay

Let

- b_1, b_2 be the current symbol received by F from the edges BF and CF, respectively,
- $d(b_n)$ be the degree of b_n, $n = 1, 2$
- and $p_c \in [0, 1]$ be a design parameter used as the probability that F performs network coding.

1) if $(d(b_1), d(b_2)) \neq (2, 2)$ {
  Change the header and transmit $b_1 \oplus b_2$ }.
 else {
  Generate a uniformly random number r_1 in the range [0,1].
  if $r_1 < p_c$ {
   Change the header and transmit $b_1 \oplus b_2$ }.
  else {
   Generate a uniformly random number r_2 from the set $\{0, 1\}$.
   if $r_2 = 0$ {
    Transmit b_1 }
   else {
    Transmit b_2 }}}
2) Repeat 1) when new b_1 and b_2 arrive.

From Step 3) in Algorithm 1, we can see that when the previous switch position is at the right buffer ($S_l = 0$), it will always be turned to the left one ($S_l = 1$) as the current symbol arrives. On the other hand, in Step 4), the switch position will only change from left to right only if this does not create the degree distribution pair we aim to avoid.

Note that the information regarding the linear combination of original symbols for each LT-coded output is contained in the header. Therefore, in Algorithm 2, the header must be changed if we transmit $b_1 \oplus b_2$.

VII. Results, Conclusions, and Future Works

Figure 9 compares the histograms of the number of symbols needed to be transmitted by the source such that all original symbols are recovered in two cases, without the proposed algorithm and with Algorithm 1. We can see that Algorithm 1 reduces the variance of the number of required symbols, thus making the histogram of the latter case more concentrated near the middle at 1270. When both algorithms are applied, as shown in Fig. 10, the histogram resembles a left-shifted version of that in Fig. 9 when only Algorithm 1 is applied. Instead of simple histograms, Figure 11 plots normalized cumulative histograms to clearly show that less symbols are needed to be transmitted when both algorithms are applied. We can conclude that our cooperative scheme improves the receiving performance, not only when the erasure probability at BD is 0.1, but also when it is 0.2 and 0.05. The larger the erasure probability, the greater the improvement.

In a more complicated network, a node can act as both a relay for upstream nodes and a source for downstream ones. Therefore, the subtree analysis as suggested in [8] is needed to identify whether a given node's buffering scheme should follow Algorithm 1, Algorithm 2, or a modified algorithm combining both of them. In addition, the relationship between erasure probabilities in all edges and the suitable parameter p_c used in Algorithm 2 should be studied further.

References

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network Information Flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204-1216, Jul. 2000.

[2] M. Luby, "LT Codes" *The* 43^{th} *Annual IEEE Symp. Foundations of Computer Science*, Nov. 2002.

[3] S. Puducheri, J. Kliewer, and T.E. Fuja, "Distributed LT Codes" *IEEE Int. Symp. Information Theory*, Jul. 2006.

[4] P. Popovski and H. Yomo, "Physical Network Coding in Two-Way Wireless Relay Channels," *IEEE Int. Conf. Communications*, Jun. 2007.

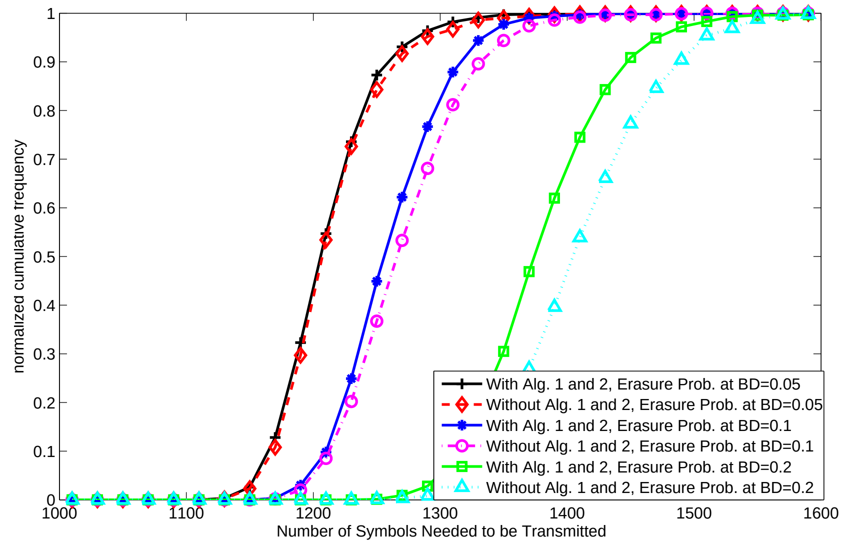

Fig. 11. Comparison of normalized cumulative histograms of the number of LT-encoded symbols needed to be transmitted in a coded butterfly network with varying erasure probability at BD such that all original symbols are recovered in two cases, without using the proposed algorithm and with Algorithms 1 and 2

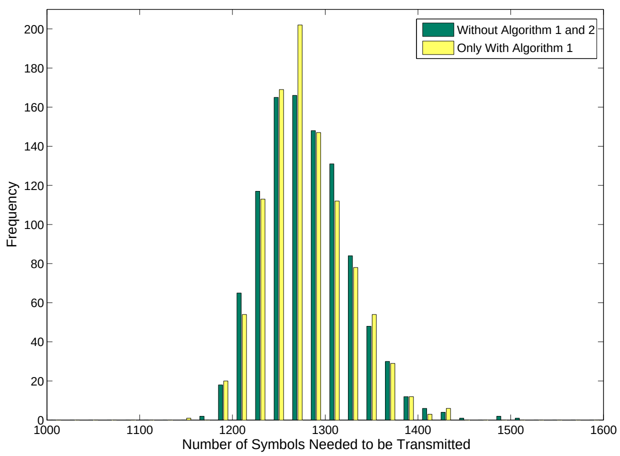

Fig. 9. Comparison of histograms of the number of LT-encoded symbols needed to be transmitted in a coded butterfly network with the erasure probability of 0.1 at BD such that all original symbols are recovered in two cases, without using the proposed algorithm and with Algorithm 1

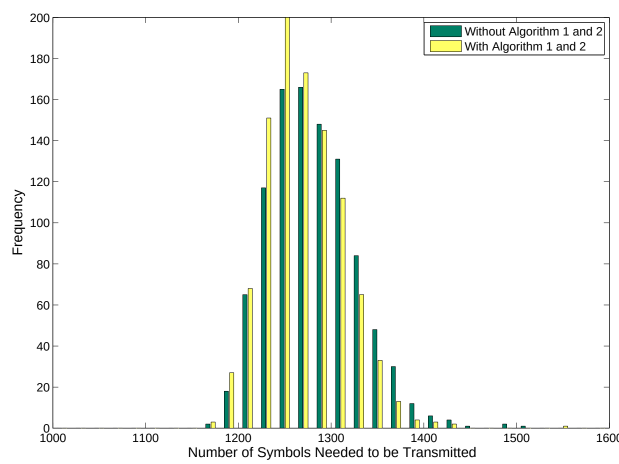

Fig. 10. Comparison of histograms of the number of LT-encoded symbols needed to be transmitted in a coded butterfly network with the erasure probability of 0.1 at BD such that all original symbols are recovered in two cases, without using the proposed algorithm and with Algorithms 1 and 2

[5] Y. Wu, P.A. Chou, and S.-Y Kung, "Information Exchange in Wireless Networks with Network Coding and Physical-Layer Broadcast" *2005 Conf. Information Sciences and Systems*, Mar. 2005.
[6] C. Hausl and J. Hagenauer, "Iterative Network and Channel Decoding for the Two-Way Relay Channel" *IEEE Int. Conf. Communications*, Jun. 2006.
[7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding" *IEEE/ACM Trans. Networking*, vol. 16, no. 3, pp. 497-510, Jun. 2008.
[8] C. Fragouli, E. Soljanin, A. Shokrollahi, "Network Coding as a Coloring Problem," in *Proc. Conf. Information Sciences and Systems.* Princeton, NJ, Mar. 2004.