Combination Network Coding: Alphabet Size and Zigzag Decoding

Chi Wan Sung
Department of Electronic Engineering
City University of Hong Kong
Email: albert.sung@cityu.edu.hk

Xueqing Gong
Department of Electronic Engineering
City University of Hong Kong
Email: xgong6-c@my.cityu.edu.hk

***Abstract*—Combination network coding can be regarded as a generalization of Maximum Distance Separable (MDS) code. An existing bound on the required alphabet size for MDS code is generalized for combination network coding. Besides, a class of combination network code called Zigzag-Decodable (ZD) code is considered. It involves only exclusive-OR and bit-shifting operations and can be decoded by a fast algorithm called zigzag decoding. It was proved that the ZD code has lower encoding and decoding complexities than other existing codes, at the expense of slight rate loss.**

I. Introduction

We consider network coding for a special class of networks, called *combination network* [1]. It is a three-layer directed acyclic network. In network coding theory, it is well known that for directed acyclic multicast problems, the cut-set bound can be achieved by linear network code, provided that the size of the finite field is large enough [2]. The combination network is often used an example to show the necessity of large field size. Besides, it is also used to show that network coding gain can be unbounded [1], [3].

Formally, the $\binom{n}{k}$ combination network is a directed acyclic network which consists of three layers of nodes, as shown in Figure 1. The top layer consists of only the source node, denoted by S. There is a directed edge from S to each of the n relay nodes in the middle layer. The bottom layer consists of $\binom{n}{k}$ sink nodes, each of which has k incoming edges connected to a distinct k-subset of relay nodes.

The source S wants to send to all the sink nodes an k-dimensional message vector $\boldsymbol{x}$, whose components are elements of a source alphabet Σ. For each edge, a symbol from an edge alphabet, Λ, can be sent in each channel use. Because of the network topology, coding can be performed only at the source S. For $i = 1, 2, \ldots, n$, let $f_i : \Sigma^k \to \Lambda$ be the encoding function for edge i which connects S to the i-th relay node. For $j = 1, 2, \ldots, \binom{n}{r}$, let $g_j : \Lambda^k \to \Sigma^k$ be the decoding function of sink j. An (n, k) combination network (CN) code is defined by these encoding and decoding functions, provided that the value of g_j is equal to $\boldsymbol{x}$ for all j and all $\boldsymbol{x} \in \Sigma^k$.

Definition 1. *The rate r of a network code for the $\binom{n}{k}$ combination network is defined as*

$$r \triangleq \frac{\log_2 |\Sigma|}{\log_2 |\Lambda|} = \frac{\log_2 q}{\log_2 q'},$$

where $q \triangleq |\Sigma|$ and $q' \triangleq |\Lambda|$.

This work was partially supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

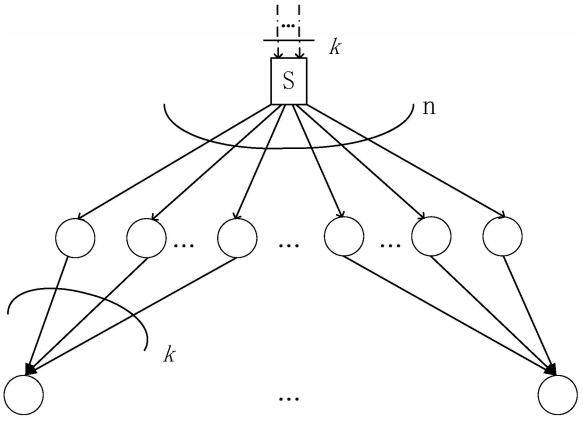

Fig. 1. The $\binom{n}{k}$ combination network.

In the $\binom{n}{k}$ combination network, the min-cut between any sink node and the source S is clearly equal to $k \log_2 q'$ bits. By the Max-flow Min-cut Theorem [4], any CN code must satisfy $k \log_2 q \leq k \log_2 q'$, or simply $q \leq q'$. In other words, its code rate r is no more than 1.

Definition 2. *An (n, k) CN code is said to be maximum distance separable (MDS) if $q = q'$.*

Note that our definition above is equivalent to the standard definition of MDS codes. In our definition, for every sink to successfully distinguish two different messages, at least $d \triangleq n - k + 1$ of the encoding functions must have different values for these two messages, meaning that the Singleton bound is satisfied with equality. In other words, MDS code in classical algebraic coding theory can be regarded as a special case of CN code [5].

II. Alphabet Size for $k = 2$

In this section, we generalize a result concerning alphabet size in network coding in [6, Chapter 2]. Let $f_1, f_2, \ldots, f_n$ be functions mapping Σ^2 to Λ. They are said to be pairwise

independent if for any pair f_i and f_j, there is an inverse function $g_{i,j} : \Lambda^2 \to \Sigma^2$ such that

$$g_{i,j}(f_i(\boldsymbol{x}), f_j(\boldsymbol{x})) = \boldsymbol{x}, \tag{1}$$

for any $\boldsymbol{x} \in \Sigma^2$.

Lemma 1. *If $f_1, f_2, \ldots, f_n$ are pairwise independent functions of the form $f_i : \Sigma^2 \to \Lambda$ and $q \le q' < q^2$, then*

$$n \le \frac{q'(q^2-1)}{q^2-q'}. \tag{2}$$

Proof: First, we claim that for each function f_i, every element in its co-domain is mapped from at most q' elements in its domain. Suppose the claim is false. Then f_i must map to the same point in Λ from more than q' points in Σ^2. By the pigeonhole principle, the function f_j, where $j \neq i$, must take on the same value for two of those points, contradicting the assumption that the functions are pairwise independent.

Now define an agreement of the function f_i to be a pair of distinct points in Σ^2 at which f_i takes the same value. Consider an arbitrary function f' equals f_i for some i. For $j = 1, 2, \ldots q'$, let M_j be the number of points in Σ^2 that map to the j-th point in Λ under f'. As shown above, we must have $0 \le M_i \le q'$. Furthermore, $\sum_{i=1}^{q'} M_i = q^2$, since there are q^2 points in the domain of f'. Therefore, f' has at least

$$\min_{M_1, M_2, \ldots, M_{q'}} \sum_{i=1}^{q'} \binom{M_i}{2}$$

agreements. It can be shown that $\sum_i M_i(M_i-1)/2$ is a Schur-convex function of M_i's, implying that the minimum is achieved when $M_1 = M_2 = \cdots = M_{q'} = q^2/q'$. In other words, f' has at least

$$\frac{q^2}{2}\left(\frac{q^2}{q'} - 1\right)$$

agreements. Totally, there are n functions. Again by the pigeonhole principle, there are at least two functions that share the same agreement if

$$\frac{nq^2}{2}\left(\frac{q^2}{q'} - 1\right) > \frac{q^2(q^2-1)}{2} \tag{3}$$

$$n > \frac{q'(q^2-1)}{q^2-q'} \tag{4}$$

Since the functions are pairwise independent, by definition, no two functions can share the same agreement, and the statement follows. ■

Theorem 2. *An $(n, 2)$ CN code must satisfy*

$$n \le \frac{q'(q^2-1)}{q^2-q'}. \tag{5}$$

In the special case when $q = q'$,

$$n \le q + 1, \tag{6}$$

and the bound is tight.

Proof: It is clear that for each sink node to be able to decode the two messages, the edges from the source to the n relay nodes must be pairwise independent. By Lemma 1, (5) must hold.

When $q = q'$, the inequality in (5) can be simplified to (6) by simple algebraic manipulation. This bound is tight because it can be achieved by a linear code over GF(q) as follows: For $i = 1, 2, \ldots, n$, let $f_i(\boldsymbol{x}) = \boldsymbol{c}_i^T \boldsymbol{x}$, where $\boldsymbol{c}_i$'s are distinct elements of the set $\{(0,1), (1,0), (1,\alpha), (1,\alpha^2), \ldots, (1,\alpha^{q-1})\}$ and α is a primitive element of GF(q). ■

For general network coding, an example network has been constructed in [6, Chapter 2] to show that a smaller alphabet size can be used if the network is operated slightly below capacity. We remark that the same phenomenon can be observed in combination network. Consider the $\binom{6}{2}$ network. To operate at full capacity, a rate-1 code is needed. By Theorem 2, $q = q' \ge 5$. Now we show that a CN code with $q = 2$ and $q' = 3$ exists. We need to define $f_i : \Sigma^2 \to \Lambda$ for $i = 1, 2, \ldots, 6$. Let $\Sigma^2 = \{a, b, c, d\}$ and $\Lambda = \{\alpha, \beta, \gamma\}$. For each f_i, we choose a pair of symbols from Σ^2 and map both of them to γ. The remaining two symbols in Σ^2 are mapped to α and β, respectively. For example, we may have $f_1(a) = \alpha, f_1(b) = \beta$ and $f_1(c) = f_1(d) = \gamma$. For f_j where $j \neq i$, the pair of symbols mapped to γ has to be different from the pair chosen for f_i. Since there are $\binom{4}{2} = 6$ possible choices, we can define six encoding functions in this way. It is easy to see that each sink can decode the message successfully.

III. MDS Code

In this section, we give a brief review on some MDS codes.

A. Reed-Solomon Code

In the original paper [7], the Reed-Solomon (RS) code is defined over the finite field GF(q) in the following way:

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}, \tag{7}$$

where $\boldsymbol{x}$ is a k-vector of information symbols, $\boldsymbol{y}$ is an n-vector of coded symbols and $\boldsymbol{G}$ is an $n \times k$ generator matrix which takes the form of the Vandermonde matrix defined below:

$$\boldsymbol{G} \triangleq \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_N & \alpha_N^2 & \cdots & \alpha_n^{k-1} \end{bmatrix}, \tag{8}$$

where the α_i's are distinct non-zero elements in GF(q), implying that $q \ge n+1$. Note that the Vandermonde matrix has the important feature that a square Vandermonde matrix (i.e., when $n = k$) is non-singular.

RS code can be applied to the combination network in a straightforward manner. The k messages are treated as elements of $\boldsymbol{x}$. The source node sends each element of $\boldsymbol{y}$ to each relay node. A sink node is connected to k of the relay nodes, and receives $\tilde{\boldsymbol{G}}\boldsymbol{x}$, where $\tilde{\boldsymbol{G}}$ is a square matrix obtained from $\boldsymbol{G}$ by retaining k of its rows. Since $\tilde{\boldsymbol{G}}$ is itself a Vandemonde matrix, it is invertible and the sink node can decode $\boldsymbol{x}$. If Gaussian elimination is used for decoding, then

$O(k^3)$ multiplications are needed. Encoding can be performed by multiplying the $n \times k$ generator matrix with the k-vector of information symbols, which requires $O(nk)$ multiplications over GF(q).

B. Binary Sequence Code

The binary sequence (BS) code is proposed for the combination network in [8]. It can be applied to the case where $q = q' = 2^w$, where $w+1$ is a prime number greater than or equal to n. Each symbol in Σ or Λ is treated as a binary vector of length w. In other words, the k message symbols, $x_1, x_2, \ldots, x_m$ are w-dimensional vectors over GF(2). The total number of information bits is kw. Note that w is commonly called the word size.

For any binary vector x of length l, let append(x) and remove(x) be the operations of appending a zero to the end of x to produce a vector of length $l+1$ and removing the last component from x to produce a vector of length $l-1$, respectively. Besides, let c-shift(x, l') be cyclicly shifting x by l' positions. For $j = 1, 2, \ldots, n$, the encoding function f_j is defined as

$$f_j(\boldsymbol{x}) = \sum_{i=1}^{k} \text{remove(c-shift(append}(x_i), (j-1)(i-1))). \quad (9)$$

Note that f_j is linear, and can be represented by a $w \times kw$ generator matrix, which is shown to be sparse with k ones in each row or column. Computation of f_j involves at most wk XOR's. The encoding complexity of the code is therefore $O(wkn)$.

For a sink to decode the message, it receives the coded message from k relay nodes and then inverts a $kw \times kw$ sparse matrix. It is shown in [8] that the matrix is invertible. Since there k^2w ones in the $kw \times kw$ matrix, the system of linear equations can be solved in $O(k^3w^2)$ binary operations using the method in [9].

C. Cauchy-RS Code

Cauchy-RS code is regarded as the state-of-the-art MDS code for data storage systems. It is an improvement over RS code [10], with two major modifications. First, Cauchy matrix, instead of Vandemonde matrix, is used as the generator matrix. The $n \times k$ Cauchy matrix is defined as

$$\begin{bmatrix} \frac{1}{x_1+y_1} & \frac{1}{x_1+y_2} & \cdots & \frac{1}{x_1+y_k} \\ \frac{1}{x_2+y_1} & \frac{1}{x_2+y_2} & \cdots & \frac{1}{x_2+y_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{x_n+y_1} & \frac{1}{x_n+y_2} & \cdots & \frac{1}{x_n+y_k} \end{bmatrix}, \quad (10)$$

where $\{x_1, x_2, \ldots, x_n\}$ and $\{y_1, y_2, \ldots, y_k\}$ are two subsets of GF(q) which satisfy $x_i + y_j \neq 0$ for $i = 1, 2, \ldots n$ and $j = 1, 2, \ldots k$. Furthermore, $q \triangleq 2^w$ is greater than or equal to $\max\{k, (n-k)\}$ [10], where w is the word size and is a positive integer. The Cauchy matrix has the nice property that every square sub-matrix of its is nonsingular. Hence, when it is applied to the $\binom{n}{k}$ combination network, every sink can decode the message.

The second modification is that the encoding and decoding of Cauchy-RS code is based on the matrix representation of of elements in GF(q) by $w \times w$ matrix of elements in GF(2), which allows operations in GF(q) be done by XOR's of elements in GF(2). Details can be found in [10].

It is shown in [10] that the encoding of Cauchy-RS code involves $O(k(n-k)\log^2 q)$ XOR's and the decoding involves $O(k(n-k)\log^2 q)$ XOR's and $O((n-k)^2)$ operations in GF(q).

IV. Zigzag-Decodable Code

In this section, we present the Zigzag-decodable (ZD) code proposed in [11]. It is a CN code, which has lower decoding complexity than the MDS codes mentioned in the previous section. The price to pay is that there is slight rate loss.

Assume that $q = 2^L$ and $q' = 2^{L+l}$. We use a polynomial over GF(2) of degrees L and $L+l$ to represent a symbol of Σ and of Λ, respectively. For $i = 1, 2, \ldots, k$, let message symbol i be represented by the polynomial

$$s_i(z) \triangleq s_{i,0} + s_{i,1}z + s_{i,2}z^2 + \cdots s_{i,L-1}z^{L-1}, \quad (11)$$

where $s_{i,j} \in$ GF(2). Define the column vectors $\boldsymbol{s}(z) \triangleq (s_1(z), s_2(z), \ldots, s_k(z))$ and $\boldsymbol{f}(\boldsymbol{s}(z)) \triangleq (f_1(\boldsymbol{s}(z)), f_2(\boldsymbol{s}(z)), \ldots, f_n(\boldsymbol{s}(z)))$. The encoding functions of the ZD code are given by

$$\boldsymbol{f}(\boldsymbol{s}(z)) = \begin{bmatrix} \boldsymbol{I}_k \\ \boldsymbol{B}(z) \end{bmatrix} \boldsymbol{s}(z), \quad (12)$$

where $\boldsymbol{I}_k$ is the $k \times k$ identity matrix and $\boldsymbol{B}(z)$ is a $(n-k) \times k$ matrix whose (i,j)-th entry is $z^{(i-1)(j-1)}$. Then $l = (n-k-1)(k-1)$. The code rate, r, is given by $L/(L+l) = L/(L+(n-k-1)(k-1))$, which approaches 1 when L goes to infinity.

For example, when $k = 3$ and $n = 7$, we have

$$\boldsymbol{A}(z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & z & z^2 \\ 1 & z^2 & z^4 \\ 1 & z^3 & z^6 \end{bmatrix}, \quad (13)$$

which is graphically shown in Figure 2. Clearly, l equals 6 in this case.

For the $\binom{n}{k}$ network, there are k systematic packets and $(n-k)$ parity packets. Each parity packet is constructed by performed at most $(k-1)L$ XORs. The encoding complexity is therefore $O((n-k)kL)$.

As for decoding, it was proved in [11] that the ZD code can be decoded by a low-complexity algorithm called Zigzag decoding, which is stated in Algorithm 1. Note that in the description of the algorithm, we use an array, rather than a polynomial, to represent a symbol. Its computational complexity is $O(k^2L)$ due to the two for-loops in lines 10 and 15, respectively, assuming that k parity packets are used for decoding. In general, assume that k_0 systematic packets

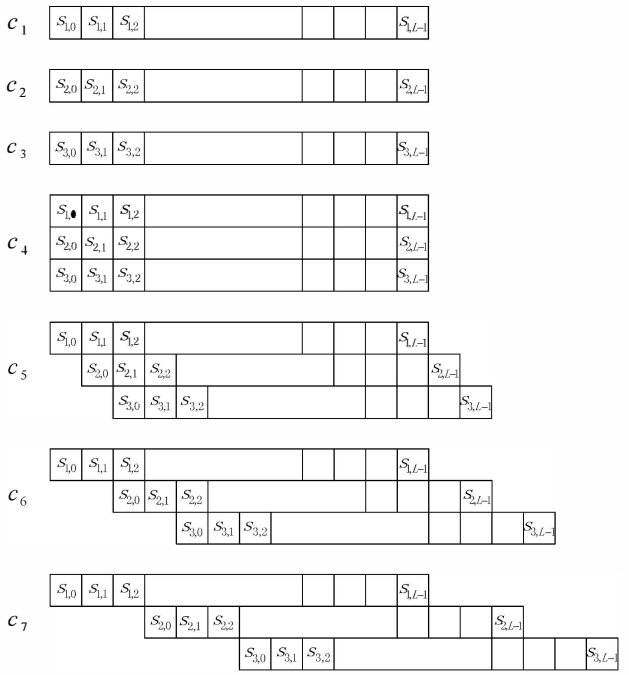

Fig. 2. ZD code with $k = 3$ and $n = 7$.

TABLE I
ENCODING AND DECODING COMPLEXITIES OF DIFFERENT CN CODES.

	Encoding	Decoding	Operations
RS	$O(nB/\log q)$	$O(k^3 + kB/\log q)$	Multiplication
BS	$O(nB)$	$O(nk^2B)$	XOR
Cauchy-RS	$O(mB\log q)$	$O(mB\log q)$ $O(m^2)$	XOR Multiplication
ZD	$O(mB)$	$O(\min\{m^2/k, k\}B)$	XOR

and k_1 packets are used for decoding, where $k_0 + k_1 = k$. The systematic packets will first be subtracted from the parity packets, which involves $O(k_1 L)$ XOR's. Decoding the parity packets involves $O(k_1^2 L)$ XOR's. Since $k_1 \le \min\{n-k, k\}$, the overall decoding complexity is $O(\min\{n-k, k\}^2 L)$.

V. PERFORMANCE COMPARISONS

A. Encoding and decoding complexities

Consider the $\binom{n}{k}$ combination network. The message size is B bits. We first analyze the encoding and decoding complexities of different coding schemes. We assume that both q and q' are powers of 2. Note that the values of q and q' are the same for all codes except for the ZD code. The message is divided into $N_b \triangleq B/(k\log_2 q)$ blocks for encoding. Each block consists of $k\log_2 q$ bits. Note that the same encoding procedure is repeated N_b times, one for each block.

For RS code, the encoding complexity is $O(nkN_b) = O(nB/\log q)$. To perform decoding, the inverse matrix can be computed in $O(k^3)$ operations and the multiplication can be carried out in $O(k^2 N_b) = O(kB/\log q)$. The overall decoding complexity is $O(k^3 + kB/\log_2 q)$. Note that all operations are performed over GF(q), where $q \ge n+1$.

For BS code, the required operations are XOR's. Its encoding complexity is $O(wknN_b) = O(nB)$ and decoding complexity is $O(k^3 w^2 N_b) = O(k^2 B\log q)$. Since q is in order of 2^n, the decoding complexity becomes $O(nk^2B)$.

For Cauchy-RS code, the encoding involves $O(k(n-k)N_b\log^2 q) = O((n-k)B\log q)$ XOR's. Its decoding involves $O(k(n-k)N_b\log^2 q) = O((n-k)B\log q)$ XOR's. Multiplications in GF(q) are needed, but they do not need to be repeated N_b times. Therefore, decoding involves $O((n-k)^2)$ operations in GF(q), where $q \ge \max\{k, n-k\}$.

For ZD code, all operations are XOR's. Its encoding complexity is $O((n-k)kLN_b) = O((n-k)B)$ and decoding complexity is $O(\min\{n-k,k\}^2 LN_b) = O(\min\{n-k,k\}^2 B/k)$.

We summarize our result in Table 1. To simplify the notation, we define $m \triangleq n-k$, which represents the number of parity packets. It is well known that the performance of RS codes suffer from the slow operations in GF(q). The other three codes work mainly over GF(2). Cauchy-RS and ZD codes involve fewer XOR operations than BS code. Since $\min\{m^2, k\} \le m$, it can be seen that the decoding complexity

Algorithm 1 ZigZag Decoding Algorithm

Input: k binary arrays, $Y_1, Y_2, \ldots, Y_k$, each of length $L+l$, and a $k \times k$ integer array, T
Output: k binary arrays, $X_1, X_2, \ldots, X_k$, each of length L

// Initialization
$B := T$;
Let V be the array of size $L+l$ whose first L elements are equal to 1 and the last r elements are equal to 0;
for i = 1 **to** k **do**
  $p_i := 1$;
  for j = 1 **to** k **do**
    Let V_j be obtained by cyclically shifting V to the right by $T[i][j]$ positions;
  end for
  $A_i := V_1 + V_2 + \cdots + V_k$;
end for
// Decoding by identifying an exposed bit in each iteration
for number of decoded bits = 1 **to** kL **do**
  Find the smallest i^* such that $A_{i^*}[p_{i^*}] = 1$;
  $j^* := \arg\min_j B[i^*][j]$;
  $b := Y_{i^*}[p_{i^*}]$ and $h := p_{i^*} - T[i^*][j^*]$;
  $X_{j^*}[h] := b$;
  // Updating variables
  for $i = 1$ **to** k **do**
    $p_i := h + T[i][j^*]$ and $Y_i[p_i] := Y_i[p_i] \oplus b$;
    $B[i][j^*] := B[i][j^*] + 1$;
    if $B[i][j^*] - T[i][j^*] - L = 0$ **then**
      $B[i][j^*] := L + l + 1$;
    end if
    if $A_i[p_i] > 1$ **then**
      $A_i[p_i] := A_i[p_i] - 1$;
    else
      $p_i := p_i + 1$;
    end if
  end for
end for

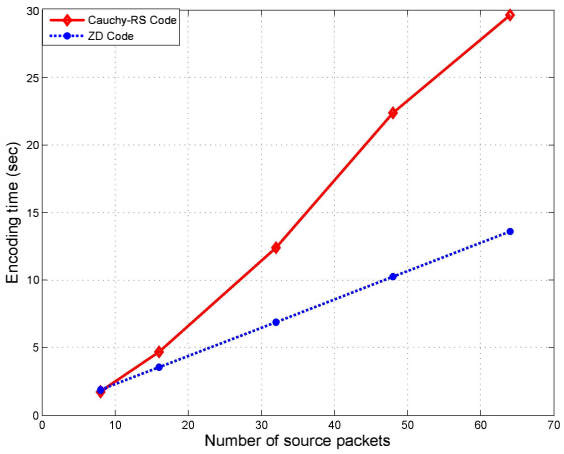

Fig. 3. Encoding time for the $\binom{n}{k}$ network with $n = 2k$.

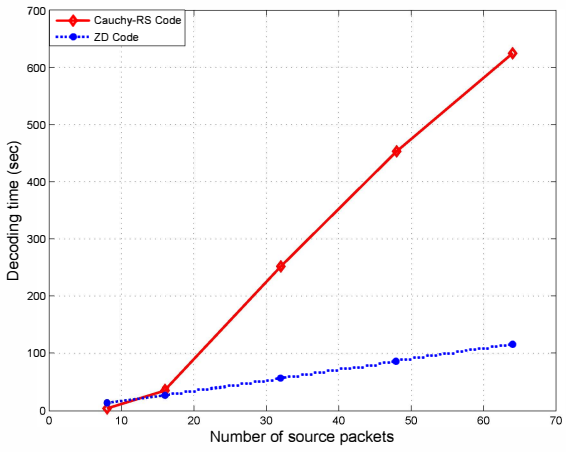

Fig. 4. Decoding time for the $\binom{n}{k}$ network with $n = 2k$.

of ZD code is slighter lower than that of Cauchy-RS.

B. Empirical results

In this section, we compare the encoding and decoding performance of Cauchy-RS Code and ZD code by experiments. The *Jerasure* library [12] is used to implement the Cauchy-RS code. The ZD code is implemented by C programming language. Our test platform is a Dell desktop with Intel Core i5-2500 CPU running at 3.30GHz with 4GB of RAM, and a L1 cache of 32KB and a L2 cache of 256KB.

The $\binom{n}{k}$ network with $n = 2k$ is considered. The value of k is chosen from $\{8, 16, 32, 64\}$. A file of size 248 Mbytes is to be encoded. As mentioned in [13], the choice of word size, w, has great influence on the performance of Cauchy-RS code, and a smaller word size often gives better performance. In the *Jerasure* library, the smallest value of w that can be chosen is $\lceil \log_2 n \rceil$, so we use this value in our experiment.

The encoding times and decoding times of the two codes are plotted in Figures 3 and 4, respectively. Each data point is obtained by the average of 10 runs. For both encoding and decoding, the computation times of both Cauchy-RS and ZD codes increase linearly but with different slopes. It can be seen that the encoding and decoding times of Cauchy-RS code increases much faster than ZD code. The encoding and decoding times of the two codes are close to each other at $k = 8$. For larger values of k, ZD code outperforms Cauchy-RS code.

VI. Conclusions

Combination network coding is considered in this paper. Our formulation allows the edge alphabet size differs from the source alphabet size. A new bound on the alphabet sizes is derived. When the two alphabet sizes are equal, the problem reduces to the classical problem of designing MDS code. Three existing MDS codes are reviewed and their encoding and decoding complexities are compared.

While MDS codes can be regarded as a CN code of rate one, we study a new CN code of rate slightly less than one, called ZD code. The main feature of this code is that it can be decoded in a very efficient manner by an algorithm called zigzag decoding using only the XOR operation. Its encoding and decoding complexities are shown to be lower than the other MDS codes we considered. We have also compared the performance of ZD code with that of Cauchy-RS code by software implementation. Numerical results show that ZD code outperforms Cauchy-RS code significantly both in terms of encoding and decoding.

References

[1] C. K. Ngai and R. W. Yeung, "Network coding gain of combination networks," in *Proc. IEEE Info. Theory Workshop*, San Antonio, Texas, Oct. 2004, pp. 283–287.

[2] R. S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE J. on Selected Areas in Commun.*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[3] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proc. ACM Symp. Parallelism in Alg. and Architect.*, San Diego, California, Jun. 2003, pp. 286–294.

[4] L. K. Ford Jr. and D. R. Fulkerson, *Flows in Network*. Princeton University Press, 1962.

[5] R. W. Yeung, S.-Y. R. Li, and N. Cai, "Network coding theory part 1: Single source," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 4, Jun. 2005.

[6] A. R. Lehman, "Network coding," Ph.D. dissertation, Massachusetts Institute of Technology, Feb 2005.

[7] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[8] M. Xiao, M. Medard, and T. Aulin, "A binary coding approach for combination networks and general erasure networks," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 786–790.

[9] D. H. Wiedemann, "Solving sparse linear equations over finite fields," *IEEE Trans. Inf. Theory*, vol. 1, no. 32, pp. 54–62, Jan. 1986.

[10] J. Blomer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman, "An XOR-based erasure-resilient coding scheme," International Computer Science Institute, Tech. Rep. TR-95-048, Aug. 1995.

[11] C. W. Sung and X. Gong, "A zigzag-decodable code with the MDS property for distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 341–345.

[12] J. S. Plank, S. Simmerman, and C. D. Schuman, "Jerasure: A library in c/c++ facilitating erasure coding for storage applications," Tech. Rep., 2007.

[13] J. S. Plank, J. Luo, C. D. Schuman, L. Xu, and Z. Wilcox-O'Hearn, "A performance evaluation and examination of open-source erasure coding libraries for storage," in *Proccedings of the 7th conference on File and storage technologies*, ser. FAST '09. Berkeley, CA, USA: USENIX Association, 2009, pp. 253–265.