Effects of the Generation Size and Overlap on Throughput and Complexity in Randomized Linear Network Coding

Yao Li, *Student Member, IEEE,* Emina Soljanin, *Senior Member, IEEE,* and Predrag Spasojević, *Member, IEEE,*

***Abstract*—To reduce computational complexity and delay in randomized network coded content distribution, and for some other practical reasons, coding is not performed simultaneously over all content blocks, but over much smaller, possibly overlapping subsets of these blocks, known as generations. A penalty of this strategy is throughput reduction. To analyze the throughput loss, we model coding over generations with random generation scheduling as a coupon collector's brotherhood problem. This model enables us to derive the expected number of coded packets needed for successful decoding of the entire content as well as the probability of decoding failure (the latter only when generations do not overlap) and further, to quantify the tradeoff between computational complexity and throughput. Interestingly, with a moderate increase in the generation size, throughput quickly approaches link capacity. Overlaps between generations can further improve throughput substantially for relatively small generation sizes.**

***Index Terms*—network coding, rateless codes, coupon collector's problem**

I. Introduction

A. Motivation: Coding over Disjoint and Overlapping Generations

Random linear network coding was proposed in [1] for "robust, distributed transmission and compression of information in networks". Subsequently, the idea found a place in a peer-to-peer(P2P) file distribution system Avalanche [2] from Microsoft. In P2P systems such as BitTorrent, content distribution involves fragmenting the content at its source, and using swarming techniques to disseminate the fragments among peers. Systems such as Avalanche, instead, circulate linear combinations of content fragments, which can be generated by any peer. The motivation behind such a scheme is that, it is hard for peers to make optimal decisions on the scheduling of fragments based on their limited local vision, whereas when fragments are linearly combined at each node, topology diversity is implanted inherently in the data flows and can be exploited without further co-ordination.

The introduction of network coding in P2P content distribution systems brings about the issue of computational complexity. Consider distributing a file consisting of N fragments, each made up of d symbols from a Galois field $GF(q)$ of size q. It takes $\mathcal{O}(Nd)$ operations in $GF(q)$ to form a linear combination per coded packet, and $\mathcal{O}(N^3 + N^2 d)$ operations, or, equivalently, $\mathcal{O}(N^2 + Nd)$ operations per information packet, to decode the information packets by solving linear equations. According to the implementers of UUSee [3], a peer-to-peer video streaming application using randomized linear coding, even with the most optimized implementation, going beyond 512 fragments in each generation risks taxing a low-end CPU, typically used in power-efficient notebook computers.

In an effort to reduce computational complexity, information packets are partitioned into disjoint subsets referred to as *generations*, and coding is done only within generations. This approach scales down the encoding and decoding problem from the whole file size N to the generation size times the number of generations. The concept of generation in network coding was first proposed by Chou et al. in [4] to handle the issue of network synchronization. Coding over randomly scheduled generations was first theoretically analyzed by Maymounkov et al. in [5]. Random scheduling of generations provides the "rateless" property which reduces the need for receiver feedback and offers resilience to various erasure patterns over the communication link. In addition, in the peer-to-peer content distribution setting, random scheduling is to some degree a good approximation when global co-ordination among peers is impractical.

With random scheduling of generations, coded packets accumulate faster in some generations than in others, even if all generations are scheduled equally probably. While waiting for the last generation to become decodable, redundant packets are accumulated in other generations. The situation is aggravated as the generation size decreases. One way to recover some of the throughput loss due to random scheduling without losing the benefits of reduced generation sizes is to allow generations to help each other in decoding. If the generations are allowed to overlap, after some of the "faster" generations are decoded, the number of unknown variables can be reduced in those generations sharing information packets with the decoded ones, which in turn reduces the number of coded packets needed to decode those generations, and enhances the throughput as a result. Our goal is to characterize the effects of generation size and overlaps on the throughput and complexity

Manuscript received April 15, 2010; revised August 14, 2010 and November 5, 2010. The material in this work was presented in part at the IEEE International Symposium on Information Theory (ISIT'10), Austin, Texas, June 2010, and the IEEE International Symposium on Network Coding (NetCod'10), Toronto, Canada, June 2010. This work was in part supported by NSF CNS Grant No. 0721888.

Y. Li and P. Spasojević are with WINLAB, the Department of Electrical and Computer Engineering, Rutgers University, North Brunswick, NJ 08902 USA emails: {yaoli,spasojev}@winlab.rutgers.edu.

E. Soljanin is with the Mathematics of Networking and Communication Department, Enabling Computing Technologies, Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974, email: emina@alcatel-lucent.com.

in randomized linear network coding.

B. Related Work

The performance of codes with random scheduling of disjoint generations was first theoretically analyzed in [5] by Maymounkov et al., who referred to them as *chunked codes*. Chunked codes allow convenient encoding at intermediate nodes, and are readily suitable for peer-to-peer file dissemination. In [5], the authors used an adversarial schedule as the network model and characterized the code performance under certain restrictions on the chunk(generation) size when the length of the information to be encoded tends to infinity.

Coding with overlapping generations was first studied in [6] and [7] with the goal to improve throughput. Reference [7] studied a "head-to-toe" overlapping scheme in which only contiguous generations overlap for a given number of information packets, and analyzed its asymptotic performance over a line network when the length of information goes to infinity. Another overlapping scheme with a grid structure was proposed in [6], analyzed for short lengths (e.g., 4 generations) and simulated for practical lengths. When properly designed, these codes show improved performance over codes with disjoint generations. In our work, we offer an analysis of coding over disjoint and overlapping generations for finite but practically long information lengths.

C. Organization and Main Contribution

In this work, coding with both disjoint and overlapping generations together with *random generation scheduling* is studied from a coupon collection [8] perspective. Previously existing results from the classical coupon collector's problem, along with our extensions, enable us to characterize the code performance with finite information lengths, from which the asymptotic code performance can further be deduced.

Section II introduces the general model for coding over generations, disjoint or overlapping, over a unicast (binary erasure) link, and characterizes the computational cost for encoding and decoding.

Section III derives several results concerning linear independence among coded packets from the same generation. Such results serve to link coupon collection to the decoding of content that has been encoded into multiple generations. Included (Claim 1) is a very good upper bound on the distribution of the number of coded packets needed for a specific generation for successful decoding.

Section IV introduces the coupon collector's brotherhood problem and its variations that can be used to model coding over generations. Probability generating functions (Theorems 2 and 4) and moments (Corollaries 3 and 5) of the number of samplings needed to collect multiple copies of distinct coupons are derived for the random sampling of a *finite* set of coupons in Section IV-A. Relevant asymptotic results on expected values and probability distributions in existing literature are recapitulated in Section IV-B for performance characterization of coding over generations in the later part of the work. The section is presented in the coupon collection language and is in itself of independent interest for general readers interested in coupon collecting problems.

In Sections V and VI, results from the previous two sections are combined to enable the analysis of the effects of generation size and overlaps on the decoding latency/throughput of coding over disjoint or overlapping generations.

Section V studies the effects of generation size on the code throughput over a BEC channel for coding over disjoint generations. Section V-A characterizes the mean and variance of the decoding latency (the number of coded packets transmitted until successful decoding) for *finite information lengths*, and Section V-B provides a lower bound for the probability of decoding failure. A large gain in throughput is observed when the generation size increases from 1 to a few tens.

In Section VI, the *random annex code* is proposed as an effort to improve code throughput by allowing random overlaps among generations. Section VI-C lists an algorithm providing precise estimation of the expected decoding latency of the random annex code. The algorithm is based on the analysis of the overlapping structure in Section VI-B and the results from the extended collector's brotherhood in Section IV. Section VI-D demonstrates the effects of overlap sizes on code throughput is shown through both numerical computation and simulations. One of our interesting observations is that overlaps between generations can provide a tradeoff between computational complexity and decoding latency. In addition, without increasing the generation size (and hence computational complexity), it is still possible to improve code throughput significantly by allowing overlaps between generations.

II. Coding Over Generations: The General Model

In this section, we describe a general random coding scheme over generations. Generations do not have to be disjoint or of equal size, and random scheduling of generations does not have to be uniform. We describe the coding scheme over a unicast link.

A. Forming Generations

The file being distributed $\mathcal{F}$ is represented as a set of N information packets, $p_1, p_2, \dots, p_N$. Each information packet is a d-dimensional column vector of information symbols in Galois Field $GF(q)$. Generations are non-empty subsets of $\mathcal{F}$.

Suppose that n generations, $G_1, G_2, \dots, G_n$, are formed s.t. $\mathcal{F} = \cup_{j=1}^{n} G_j$. A coding scheme is said to be non-overlapping if the generations are disjoint, i.e., $\forall i \neq j$, $G_i \cap G_j = \emptyset$; otherwise, the scheme is said to be overlapping. The size of each generation G_j is denoted by g_j, and its elements $p_1^{(j)}, p_2^{(j)}, \dots, p_{g_j}^{(j)}$. For convenience, we will occasionally also use G_j to denote the matrix with columns $p_1^{(j)}, p_2^{(j)}, \dots, p_{g_j}^{(j)}$.

B. Encoding

In each transmission, the source first selects one of the n generations at random. The probability of choosing generation G_i is ρ_i, $\sum_{i=1}^{n} \rho_i = 1$. Let $\boldsymbol{\rho} = (\rho_1, \rho_2, \dots, \rho_n)$. Once generation G_j is chosen, the source chooses a coding vector

$\mathbf{e} = [e_1, e_2, \ldots, e_{g_j}]^T$, with each of the g_j components chosen independently and equally probably from $GF(q)$. A new packet $\bar{p}$ is then formed by linearly combining packets from G_j by $\mathbf{e}$: $\bar{p} = \sum_{i=1}^{g_j} e_i p_i^{(j)} = \mathbf{e} \cdot G_j$ (G_j here denotes a matrix).

The coded packet $\bar{p}$ is then sent over the communication link to the receiver along with the coding vector $\mathbf{e}$ and the generation index j. Figure 1 shows a diagram of the communication between the source and the receiver. The generations shown in this example are chosen to be disjoint, but this is not necessary.

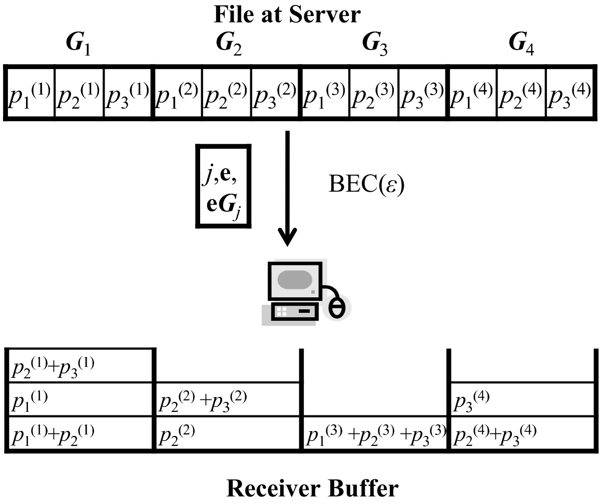


Fig. 1. A file divided into $N = 12$ fragments and $n = 4$ (disjoint) generations containing $h = 3$ fragments each is available for distribution at the server. A receiver collects random linear combinations of randomly scheduled generations.

C. Decoding

Decoding starts with any generation G_j for which the receiver has collected g_j coded packets with linearly independent coding vectors. The information packets making up this generation are decoded by solving a system of g_j linear equations in $GF(q)$ formed by the coded packets on one side and the linear combinations of the information packets by the coding vectors on the other. Since generations are allowed to overlap, a decoded information packet may also participate in other generations, from the equations of which the information packet is then removed as an unknown variable. Consequently, in all the generations overlapping with the decoded generations, the number of unknown packets is reduced. As a result, some generations may become decodable even if no new coded packets are received from the source. Again, the newly decoded generations resolve some unknowns of the generations they overlap with, which in turn may become decodable and so on. We declare successful decoding when all N information packets have been decoded.

The coding scheme described here is inherently rateless and easily extendable to more general network topologies that allow coding at intermediate network nodes.

D. Packet Overhead

Contained in each coded packet are the index of a generation G_j and a linear combining vector for G_j which together take up $\lceil \log_2 n \rceil + g_j \lceil \log_2 q \rceil$ bits. Meanwhile, the data in each coded packet comprise $d\lceil \log_2 q \rceil$ bits. The generation size makes a more significant contribution to packet overhead and such contribution is non-negligible due to the limited size ($\sim$ a few KB) of transmission packets in practical networks. This gives another reason to keep generations small, besides reducing computational complexity.

E. Computational Complexity

The computational complexity for encoding is $\mathcal{O}(d \max\{g_j\})$ per coded packet for linearly combining the g_j information packets in each generation (recall that d is the number of $GF(q)$ symbols in each information packet, as defined in Section II-A). For decoding, the largest number of unknowns in the systems of linear equations to be solved is not more than $\max\{g_j\}$, and therefore the computational complexity is upper bounded by $\mathcal{O}((\max\{g_j\})^2 + d \max\{g_j\})$ per information packet.

F. Decoding Latency

In this paper, we focus on the tradeoff between the computational complexity and the *decoding latency* of these codes over unicast links with erasures. Decoding latency here is defined as the number of coded packets transmitted until successful decoding of all the information packets, and *overhead* is the difference between the number of information packets and the decoding latency. We assume a memoryless BEC with a constant erasure rate ϵ. Since our coding scheme is rateless, each coded packet is statistically of the same importance, and so the average decoding latency is inversely proportional to the achievable capacity $(1 - \epsilon)$ of the link. The throughput of the code is inversely proportional to the decoding latency for given information length.

III. Collecting Coded Packets and Decoding

A generation G_i is not decodable until the number of linearly independent equations collected for G_i reaches the number of its information packets not yet resolved by decoding other generations. The connection between the number of coded packets collected and the linear independence among these coded packets has to be established before we can predict the decoding latency of codes over generations using the collector's brotherhood model that will be discussed in the next section.

Let $M(g, x)$ be the number of coded packets from a generation of size g adequate for collecting x linearly independent equations. Then $M(g, x)$ has expected value [9]

$$E[M(g,x)] = \sum_{j=0}^{x-1} \frac{1}{1 - q^{j-g}}. \tag{1}$$

Approximating summation by integration, from (1) we get

$$\begin{aligned} E[M(g,x)] \lessapprox & \int_0^{x-1} \frac{1}{1 - q^{y-g}} dy + \frac{1}{1 - q^{x-1-g}} \\ = & x + \frac{q^{x-1-g}}{1 - q^{x-1-g}} + \log_q \frac{1 - q^{-g}}{1 - q^{x-1-g}}. \end{aligned} \tag{2}$$

Let

$$\eta_g(x) = x + \frac{q^{x-1-g}}{1-q^{x-1-g}} + \log_q \frac{1-q^{-g}}{1-q^{x-1-g}}. \qquad (3)$$

We can use $\eta_g(x)$ to estimate the number of coded packets needed from a certain generation to gather x linearly independent equations.

In addition, we have the following Claim 1 which upper bounds the tail probability of $M(g, g)$, the number of coded packets needed for a certain generation to gather enough linearly independent equations for decoding.

Claim 1: There exist positive constants $\alpha_{q,g}$ and $\alpha_{2,\infty}$ such that, for $s \geq g$,

$$\begin{aligned}\text{Prob}[M(g,g) > s] &= 1 - \prod_{k=0}^{g-1}(1-q^{k-s})\\ &< 1 - \exp(-\alpha_{q,g}q^{-(s-g)}) < 1 - \exp(-\alpha_{2,\infty}q^{-(s-g)}).\end{aligned}$$

Also, since $1 - \exp(-x) < x$ for $x > 0$,

$$\text{Prob}[M(g,g) > s] < \alpha_{q,g}q^{-(s-g)}. \qquad (4)$$

Proof: Please refer to Appendix A. ■

We will use Claim 1 in Theorem 8 in Section V to derive an upper bound to the expected overhead of coding over disjoint generations.

IV. Coupon Collector's Brotherhood and Collecting Coded Packets from Generations

The coupon collector's brotherhood problem [10], [11] studies quantities related to the completion of m sets of n distinct coupons by sampling a set of n distinct coupons uniformly at random with replacement. In analogy, coded packets belonging to generation j can be viewed as copies of coupon j, and hence the process of collecting coded packets when generations are scheduled uniformly at random can be modeled as collecting multiple copies of distinct coupons.

Because of possible linear dependence among coded packets and the overlaps between generations, the numbers of coded packets needed for each of the n generations to ensure successful decoding, however, are n random variables. Therefore, we must generalize the coupon collector's brotherhood model from collecting a uniform number of copies for all coupons to collecting different numbers of copies for different coupons, before it can be applied to the analysis of the throughput performance of coding over generations. In this section, the original collector's brotherhood model is generalized in two ways. And later in this paper, the analysis of the throughput performance of coding over disjoint generations in Section V rests on the first generalization, whereas that of coding over overlapping generations in Section VI rests on the second generalization. As our results are of more general interest than the coding-over-generations problem, we will express them in the coupon collection language. For example, the probability ρ_i of scheduling generation G_i (defined in Section II) here refers to the probability of sampling a copy of coupon G_i, for $i = 1, 2, \ldots, n$.

A. Generating Functions, Expected Values and Variances

For any $m \in \mathbb{N}$, we define $S_m(x)$ as follows:

$$S_m(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{m-1}}{(m-1)!} \quad (m \geq 1) \qquad (5)$$

$$S_m(x) = 0 \quad (m \leq 0) \text{ and } S_\infty(x) = e^x. \qquad (6)$$

Let the total number of samplings needed to ensure that at least $m_i (\geq 0)$ copies of coupon G_i are collected for all $i = 1, 2, \ldots, n$ be $T(\boldsymbol{\rho}, \mathbf{m})$, where $\mathbf{m} = (m_1, m_2, \ldots, m_n)$. The following Theorem 2 gives $\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z)$, the generating function of the tail probabilities of $T(\boldsymbol{\rho}, \mathbf{m})$. This result is generalized from [10] and [11], and its proof uses the Newman-Shepp symbolic method in [10]. Boneh et al. [12] gave the same generalization, but we restate it here for use in our analysis of coding over disjoint generations. If for each $j = 1, 2, \ldots, n$, the number of coded packets needed from generation G_j for its decoding is known to be m_j (which can be strictly larger than the generation size g_j), $T(\boldsymbol{\rho}, \mathbf{m})$ then gives the total number of coded packets needed to ensure successful decoding of the entire content when the generations are scheduled according to the probability vector $\boldsymbol{\rho}$.

Theorem 2: (Non-Uniform Sampling) Let

$$\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z) = \sum_{i \geq 0} \text{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) > i]z^i. \qquad (7)$$

Then,

$$\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z) = \qquad (8)$$

$$\int_0^\infty \left\{e^{-x(1-z)} - \prod_{i=1}^n \left[e^{-\rho_i x(1-z)} - S_{m_i}(\rho_i xz)e^{-\rho_i x}\right]\right\}dx.$$

Proof: Please refer to Appendix B, where we give a full proof of the theorem to demonstrate the Newman-Shepp symbolic method [10], which is also used in the proof of our other generalization in Theorem 4. ■

The expected value and the variance of $T(\boldsymbol{\rho}, \mathbf{m})$ follow from the tail probability generating function derived in Theorem 2.

Corollary 3:

$$\begin{aligned}E[T(\boldsymbol{\rho}, \mathbf{m})] &= \varphi_{T(\boldsymbol{\rho},\mathbf{m})}(1)\\ &= \int_0^\infty \left\{1 - \prod_{i=1}^n \left[1 - S_{m_i}(\rho_i x)e^{-\rho_i x}\right]\right\}dx,\\ Var[T(\boldsymbol{\rho}, \mathbf{m})] &= 2\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(1) + \varphi_{T(\boldsymbol{\rho},\mathbf{m})}(1) - \varphi^2_{T(\boldsymbol{\rho},\mathbf{m})}(1).\end{aligned}$$

Proof: Please refer to Appendix B. ■

Note that in Theorem 2 and Corollary 3, m_i-s are allowed to be 0, thus including the case where only a specific subset of the coupons is of interest. Theorem 2 and Corollary 3 are also useful for the analysis of coding over generations when there is a difference in priority among the generations. For instance, in layered coded multimedia content, the generations containing the packets of the basic layer could be given a higher priority than those containing enhancement layers because of a hierarchical reconstruction at the receiver.

In the following, we present another generalization of the collector's brotherhood model. Sometimes we are simply interested in collecting a coupon subset of a certain size, regardless of the specific content of the subset. This can be further extended to the following more complicated case: for each $i = 1, 2, \ldots, A (A \geq 1)$, ensure that there exists a subset of $\{G_1, G_2, \ldots, G_n\}$ such that each of its k_i elements has at least m_i copies in the collected samples. Such a generalization is intended for treatment of coding over equally important generations, for example, when each generation is a substream of multiple-description coded data. In this generalization, the generation scheduling (coupon sampling) probabilities are assumed to be uniform, i.e., $\rho_1 = \rho_2 = \cdots = \rho_n = 1/n$.

Suppose that for some positive integer $A \leq n$, integers $k_1, \ldots, k_A$ and $m_1, \ldots, m_A$ satisfy $1 \leq k_1 < \cdots < k_A \leq n$ and $\infty = m_0 > m_1 > \cdots > m_A > m_{A+1} = 0$. We are interested in the total number $U(\mathbf{m}, \mathbf{k})$ of coupons that needs to be collected, to ensure that the number of distinct coupons for which at least m_i copies have been collected is at least k_i, for all $i = 1, 2, \ldots, A$, where $\mathbf{m} = (m_1, m_2, \ldots, m_A)$ and $\mathbf{k} = (k_1, k_2, \ldots, k_A)$. The following Theorem 4 gives the generating function $\varphi_{U(\mathbf{m},\mathbf{k})}(z)$ of $U(\mathbf{m}, \mathbf{k})$.

Theorem 4: (Uniform Sampling)

$$\varphi_{U(\mathbf{m},\mathbf{k})}(z) = n \int_0^\infty e^{-nx} \Big\{ e^{nxz} - \tag{9}$$

$$\sum_{\substack{(i_0, i_1, \ldots, i_{A+1}): \\ i_0 = 0, i_{A+1} = n \\ i_j \in [k_j, i_{j+1}] \\ j = 1, 2, \ldots, A}} \prod_{j=0}^{A} \binom{i_{j+1}}{i_j} \Big[S_{m_j}(xz) - S_{m_{j+1}}(xz) \Big]^{i_{j+1} - i_j} \Big\} dx.$$

Proof: Please refer to Appendix B. ■

Same as for Corollary 3, we can find $E[U(\mathbf{m}, \mathbf{k})] = \varphi_{U(\mathbf{m},\mathbf{k})}(1)$. A computationally wieldy representation of $E[U(\mathbf{m}, \mathbf{k})]$ is offered in the following Corollary 5 in a recursive form.

Corollary 5: For $k = k_1, k_1 + 1, \ldots, n$, let

$$\phi_{0,k}(x) = [(S_{m_0}(x) - S_{m_1}(x))e^{-x}]^k;$$

For $j = 1, 2, \ldots, A$, let

$$\phi_{j,k}(x) = \sum_{w=k_j}^{k} \binom{k}{w} \left[(S_{m_j}(x) - S_{m_{j+1}}(x))e^{-x} \right]^{k-w} \phi_{j-1,w}(x),$$

for $k = k_{j+1}, k_{j+1} + 1, \ldots, n$.

Then,

$$E[U(\mathbf{m}, \mathbf{k})] = n \int_0^\infty (1 - \phi_{A,n}(x))\, dx. \tag{10}$$

It is not hard to find an algorithm that calculates $1 - \phi_{A,n}(x)$ in $(c_1 m_1 + c_2(n-1) + c_3 \sum_{j=1}^{A} \sum_{k=k_{j+1}}^{n} (k - k_j))$ basic arithmetic operations, where c_1, c_2 and c_3 are positive constants. As long as $m_1 = \mathcal{O}(An^2)$, we can estimate the amount of work for a single evaluation of $1 - \phi_{A,n}(x)$ to be $\mathcal{O}(An^2)$. The integral (10) can be computed through the use of an efficient quadrature method, for example, Gauss-Laguerre quadrature. For reference, some numerical integration issues for the special case where $A = 1$ have been addressed in Part 7 of [13] and in [12].

In Section VI, we will apply Corollary 5 to find out the expected throughput of the *random annex code*, an overlapping coding scheme in which generations share randomly chosen information packets. The effect of the overlap size on the throughput can be investigated henceforth.

B. Limiting Mean Value and Distribution

In the previous subsection, we considered collecting a finite number of copies of a coupon set of a finite size. In this part, we present some results from existing literature on the limiting behavior of $T(\boldsymbol{\rho}, \mathbf{m})$ as $n \to \infty$ or $m_1 = m_2 = \cdots = m_n = m \to \infty$, assuming $\rho_1 = \rho_2 = \cdots = \rho_n = \frac{1}{n}$. By slight abuse in notation, we denote $T(\boldsymbol{\rho}, \mathbf{m})$ here as $T_n(m)$.

By Corollary 3,

$$E[T_n(m)] = n \int_0^\infty \left[1 - (1 - S_m(x)e^{-x})^n \right] dx. \tag{11}$$

The asymptotics of $E[T_n(m)]$ for large n has been discussed in literature [10], [14] and [15], and is summarized in the following Theorem 6, (13), and Theorem 7.

Theorem 6: ([14]) When $n \to \infty$,

$$E[T_n(m)] = n \log n + (m-1) n \log\log n + C_m n + o(n), \tag{12}$$

where $C_m = \gamma - \log(m-1)!$, γ is Euler's constant, and $m \in \mathbb{N}$.

For $m \gg 1$, on the other hand, we have [10]

$$E[T_n(m)] \to nm. \tag{13}$$

What is worth mentioning is that, as the number of coupons $n \to \infty$, for the first complete set of coupons, the number of samplings needed is $\mathcal{O}(n \log n)$, whereas the additional number of samplings needed for each additional set is only $\mathcal{O}(n \log\log n)$.

In addition to the expected value of $T_n(m)$, the concentration of $T_n(m)$ around its mean is also of great interest to us. This concentration leads to an estimate of the probability of successful decoding for a given number of collected coded packets. We can specialize Corollary 3 to derive the variance of $T_n(m)$, as a measure of probability concentration.

Further, since the tail probability generating functions derived in the last subsection are power series of non-negative coefficients and are convergent at 1, they are absolutely convergent on and inside the circle $|z| = 1$ in the complex z-plane. Thus, it is possible to compute the tail probabilities using Cauchy's contour integration formula. However, extra care is required for numerical stability in such computation.

Here we instead look at the asymptotic case where the number of coupons $n \to \infty$. Erdös and Rényi have proven in [16] the limit law of $T_n(m)$ as $n \to \infty$. Here we restate Lemma B from [14] by Flatto, which in addition expresses the rate of convergence to the limit law. We will later use this result to derive a lower bound for the probability of decoding failure in Theorem 9 in Section V-B.

Theorem 7: ([14]) Let

$$Y_n(m) = \frac{1}{n} \left(T_n(m) - n \log n - (m-1) n \log\log n \right).$$

Then,

$$\Pr[Y_n(m) \leq y] = \exp\left(-\frac{e^{-y}}{(m-1)!} \right) + \mathcal{O}\left(\frac{\log\log n}{\log n} \right).$$

Remark 1: (Remarks 2&3, [14]) The estimation in Theorem 7 is understood to hold uniformly on any finite interval $-a \leq y \leq a$. i.e., for any $a > 0$,

$$\left| \text{Prob}\left[Y_n(m) \leq y\right] - \exp\left(-\frac{\exp(-y)}{(m-1)!}\right) \right| \leq C(m,a)\frac{\log\log n}{\log n},$$

$n \geq 2$ and $-a \leq y \leq a$. $C(m,a)$ is a positive constant depending on m and a, but independent of n. For $m = 1$, the convergence rate to limit law is much faster: the $\mathcal{O}\left(\frac{\log\log n}{\log n}\right)$ term becomes $\mathcal{O}\left(\frac{\log n}{n}\right)$.

V. Coding Over Disjoint Generations

In this section, we study the performance of coding over disjoint generations. We derive both an upper bound and a lower bound for the expected decoding latency (as defined in Section II-F). We also derive the variance of the decoding latency.

A. Expected Decoding Latency and Its Variance

Let M_i $(i = 1, 2, \ldots, n)$ be the number of collected coded packets from generation G_i when G_i first becomes decodable. Then M_i is at least g_i, has the same distribution as $M(g_i, g_i)$, the number of coded packets needed for a certain generation to gather enough linearly independent equations for decoding, as defined and studied in Section III. M_i's are independent random variables. Let the decoding latency over a perfect channel be $W(\boldsymbol{\rho}, \mathbf{g})$, where $\mathbf{g} = (g_1, g_2, \ldots, g_n)$. Use $W_\epsilon(\boldsymbol{\rho}, \mathbf{g})$ to denote the decoding latency on a BEC(ϵ).

Let X_k $(k = 1, 2, \ldots)$ be i.i.d. geometric random variables with success rate $1-\epsilon$. Therefore, $E[X_k] = \frac{1}{1-\epsilon}$ and $E[X_k^2] = \frac{1+\epsilon}{(1-\epsilon)^2}$. Then

$$W_\epsilon(\boldsymbol{\rho}, \mathbf{g}) = \sum_{i=1}^{W(\boldsymbol{\rho},\mathbf{g})} X_i,$$

and therefore,

$$E[W_\epsilon(\boldsymbol{\rho}, \mathbf{g})] = \frac{1}{1-\epsilon}E[W(\boldsymbol{\rho}, \mathbf{g})], \tag{14}$$

$$Var[W_\epsilon(\boldsymbol{\rho}, \mathbf{g})] = \frac{1}{(1-\epsilon)^2}\left(Var[W(\boldsymbol{\rho}, \mathbf{g})] + \epsilon E[W^2(\boldsymbol{\rho}, \mathbf{g})]\right). \tag{15}$$

By definition, $E[W(\boldsymbol{\rho}, \mathbf{g})]$ is lower bounded by $E[T(\boldsymbol{\rho}, \mathbf{g})]$, the expected number of coded packets necessary for collecting at least g_i coded packets for each generation G_i, and $E[T(\boldsymbol{\rho}, \mathbf{g})]$ is as given in Corollary 3.

The following Theorem 8 gives the exact expression for the first and second moments of $W(\boldsymbol{\rho}, \mathbf{g})$, along with an upper bound for $E[W(\boldsymbol{\rho}, \mathbf{g})]$ considering the effect of finite finite field size q. Then, the expected value and the variance of $W_\epsilon(\boldsymbol{\rho}, \mathbf{g})$ can be derived from (14) and (15).

Theorem 8: The expected number of coded packets needed for successful decoding of all N information packets

$$\begin{aligned} &E[W(\boldsymbol{\rho}, \mathbf{g})] \\ &= \int_0^\infty \left(1 - \prod_{i=1}^n \left(1 - e^{-\rho_i x} E_{M_i}\left[S_{M_i}(\rho_i x)\right]\right)\right) dx \qquad (16) \\ &< \int_0^\infty \left(1 - \prod_{i=1}^n \Big(1 - e^{-\rho_i x}\big(S_{g_i}(\rho_i x) \qquad (17) \\ &\quad + \alpha_{q,g_i} q^{g_i} e^{\rho_i x/q} - \alpha_{q,g_i} q^{g_i} S_{g_i}(\rho_i x/q)\big)\Big)\right) dx, \end{aligned}$$

$$\begin{aligned} &E[W^2(\boldsymbol{\rho}, \mathbf{g})] \qquad (18) \\ &=2\int_0^\infty x\Bigg(1 - \sum_{i=1}^n \rho_i \frac{1 - E_{M_i}[S_{M_i-1}(\rho_i x)]e^{-\rho_i x}}{1 - E_{M_i}\left[S_{M_i}(\rho_i x)\right]e^{-\rho_i x}} \cdot \\ &\quad \cdot \prod_{j=1}^n \left(1 - E_{M_j}\left[S_{M_j}(\rho_j x)\right]e^{-\rho_j x}\right)\Bigg) dx \\ &\quad + \int_0^\infty \left(1 - \prod_{i=1}^n \left(1 - e^{-\rho_i x} E_{M_i}[S_{M_i}(\rho_i x)]\right)\right) dx \end{aligned}$$

where $\alpha_{q,g_i} = -\sum_{k=0}^{g_i-1} \ln\left(1 - q^{k-g_i}\right)$, $i = 1, 2, \ldots, n$.

Proof: Please refer to Appendix C. ■

In the case where generations are of equal size and scheduled uniformly at random, we can estimate the asymptotic lower bound for $E[W(\boldsymbol{\rho}, \mathbf{g})]$ by the asymptotics of $T_n(m)$ given in (12) and (13).

Figure 2(a) shows several estimates of $E[W(\boldsymbol{\rho}, \mathbf{g})]$, and Figure 2(b) shows the standard deviation of $W(\boldsymbol{\rho}, \mathbf{g})$ calculated from Theorem 8 and simulation results, when $\rho_i = \frac{1}{n}$ and $g_i = g$ for $i = 1, 2, \ldots, n$. The estimates are plotted versus the uniform generation size g for fixed $N = ng = 1000$.

For coding over disjoint generations and a fixed total number of information packets, both the expected value and the standard deviation of the decoding latency drop significantly as the generation size g grows to a relatively small value from the case where no coding is used ($g = 1$). Hence, throughput is improved by a moderate increase in the computational cost that scales quadratically with the generation size (see Section II-E). On the other hand, we also observe that past a moderate generation size ($\sim 50 - 100$ coded packets for $N = 1000$), the decrease in decoding latency becomes slower by further increasing the encoding/decoding complexity. We therefore argue for a "sweet spot" generation size which characterizes the tradeoff between throughput and complexity.

B. Probability of Decoding Failure

In this subsection we assume uniform generation size and scheduling probability, i.e., $\rho_i = \frac{1}{n}$, $g_i = g$ for $i = 1, 2, \ldots, n$. For short, we denote $W(\boldsymbol{\rho}, \mathbf{g})$ as $W_n(g)$. From Theorem 7, we obtain the following lower bound to the probability of decoding failure as $n \to \infty$:

Theorem 9: When $n \to \infty$, the probability of decoding failure when t coded packets have been collected is greater than $1 - \exp\left[-\frac{1}{(g-1)!}n(\log n)^{g-1}\exp\left(-\frac{t}{n}\right)\right] + \mathcal{O}\left(\frac{\log\log n}{\log n}\right)$.

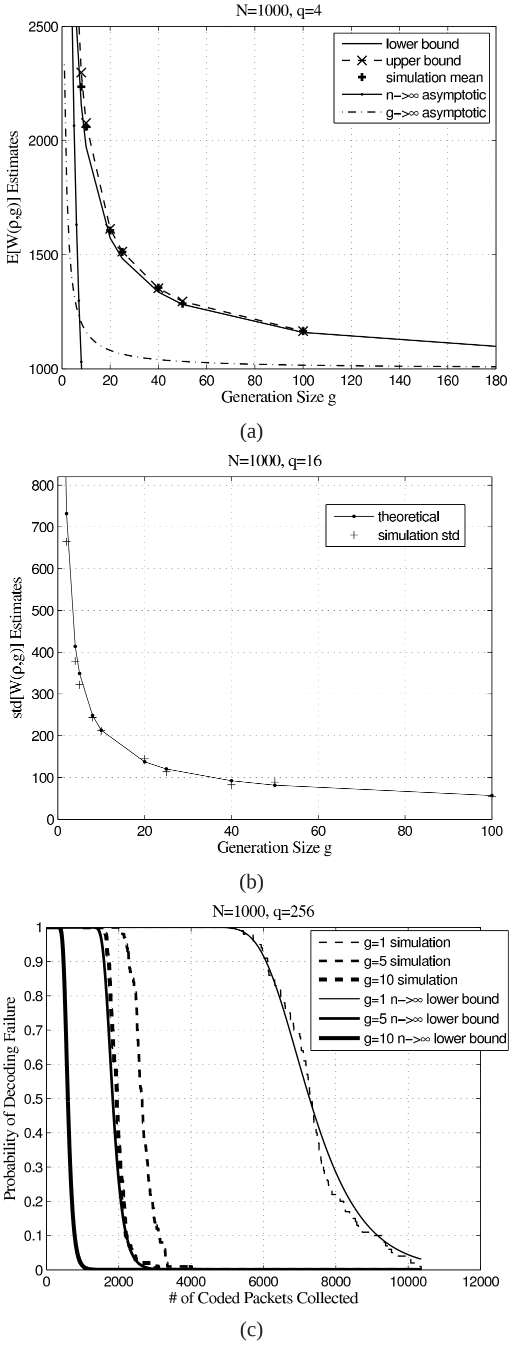


Fig. 2. (a) Estimates of $E[W(\boldsymbol{\rho}, \mathbf{g})]$, the expected number of coded packets required for successful decoding when the total number of information packets is $N = 1000$, and both $\mathbf{g}$ and $\boldsymbol{\rho}$ are uniform. Estimates shown: lower bound $E[T(\boldsymbol{\rho}, \mathbf{g})]$; upper bound (17); mean of $W(\boldsymbol{\rho}, \mathbf{g})$ in simulation; $n \to \infty$ asymptotic (12); $m \gg 1$ asymptotics (13); (b) Estimates of the standard deviation of $W(\boldsymbol{\rho}, \mathbf{g})$; (c) Estimates of probability of decoding failure versus the number of coded packets collected: Theorem 9 along with simulation results.

Proof: The probability of decoding failure after acquiring t coded packets equals $\text{Prob}[W_n(g) > t]$. Since $W_n(g) \geq T_n(g)$,

$$\begin{aligned}&\text{Prob}[W_n(g) > t] \geq \text{Prob}[T_n(g) > t]\\&= 1 - \text{Prob}\left[Y_n(g) \leq \frac{t}{n} - \log n - (g-1)\log\log n\right].\end{aligned}$$

The result in Theorem 9 follows directly from Theorem 7. ■

Corollary 10: When g is fixed and $n \to \infty$, in order to make the probability of decoding failure smaller than δ, the number of coded packets collected has to be at least $E[T_n(g)] - n \log\log \frac{1}{1-\delta}$. If $\delta = \frac{1}{N^c}$ for some constant c, then the number of coded packets necessary for successful decoding has to be at least $E[T_n(g)] + cn\log(ng)$.

Theorem 4.2 in [5] also gives the number of coded packets needed to have the probability of decoding failure below $\delta = \frac{1}{N^c}$, but under the assumption that $\ln(N/\delta) = o(N/n) = o(g)$. In comparison, Corollary 10 treats the case where g is constant.

Figure 2(c) shows the estimate of the probability of decoding failure versus T, the number of coded packets collected. As pointed out in Remark 1, for $m \geq 2$, the deviation of the CDF of $T_n(m)$ from the limit law for $n \to \infty$ depends on m and is on the order of $\mathcal{O}(\frac{\log\log n}{\log n})$ for $m \geq 2$, which is quite slow, partly explaining the deviation of the limit law curves from the simulation curves for $m = 5$ and $m = 10$ in Figure 2(c).

VI. Coding Over Overlapping Generations

Even when generations are scheduled uniformly at random, there will be more coded packets accumulated in some of the generations than in others. The "slowest" generation is the bottleneck for file decoding. It is then advisable to design a mechanism that allows "faster" generations to help those lagging behind. In this section, we propose the *random annex code*, a new coding scheme in which generations share randomly chosen packets, as opposed to previously proposed "head-to-toe" overlapping scheme of [7].

We provide a heuristic analysis of the code throughput based on our results for the coupon collection model and an examination of the overlapping structure. Previous work on coding over overlapping generations, [6] and [7], lacks accurate performance analysis for information blocks of moderate finite lengths. On the other hand, the computational effort needed to carry out our analysis scales well with the length of information, and the performance predictions coincide with simulation data. In addition, we find that our random annex code outperforms the "head-to-toe" overlapping scheme of [7] over a unicast link.

In this section we conveniently assume that the coded packets are sent over a perfect channel, since here we are interested in comparing the performance of different rateless coding schemes.

A. Forming Overlapping Generations

We form n overlapping generations out of a file with N information packets in two steps as follows:

1) Partition the file set $\mathcal{F}$ of N packets into subsets $B_1, B_2, \ldots, B_n$, each containing h consecutive packets. These $n = N/h$ subsets are referred to as *base generations*. Thus, $B_i = \{p_{(i-1)h+1}, p_{(i-1)h+2}, \ldots, p_{ih}\}$ for $i = 1, 2, \ldots, n$. N is assumed to be a multiple of h for convenience. In practice, if N is not a multiple of h, set $n = \lceil N/h \rceil$ and assign the last $[N - (n-1)h]$ packets to the last (smaller) base generation.
2) To each base generation B_i, add a random *annex* R_i, consisting of l packets chosen uniformly at random (without replacement) from the $N - h = (n-1)h$ packets in $\mathcal{F} \backslash B_i$. The base generation together with its annex constitutes the *extended generation* $G_i = B_i \cup R_i$, the size of which is $g = h + l$. Throughout this paper, unless otherwise stated, the term "generation" will refer to "extended generation" whenever used alone for overlapping generations.

The generation scheduling probabilities are chosen to be uniform, $\rho_1 = \rho_2 = \cdots = \rho_n = 1/n$. The encoding and decoding procedures run the same as described in the general model in Section II.

B. Analyzing the Overlapping Structure

The following Claims 11 through 14 present combinatorial derivations of quantities concerning the frequency at which an arbitrary information packet is represented in different generations.

Claim 11: For any packet in a base generation B_k, the probability that it belongs to annex R_r for some $r \in \{1, 2, \ldots, n\} \backslash \{k\}$ is

$$\pi = \binom{N-h-1}{l-1} / \binom{N-h}{l} = \frac{l}{N-h} = \frac{l}{(n-1)h},$$

whereas the probability that it does not belong to R_r is $\bar{\pi} = 1 - \pi$.

Claim 12: Let X be the random variable representing the number of generations an information packet participates in. Then, $X = 1 + Y$, where Y is $\mathrm{Binom}(n-1, \pi)$.

$$E[X] = 1 + (n-1)\pi = 1 + \frac{l}{h},$$

and

$$Var[X] = (n-1)\pi\bar{\pi}.$$

Claim 13: In each generation of size $g = h + l$, the expected number of information packets not participating in any other generation is $h\bar{\pi}^{(n-1)} \approx he^{-l/h}$ for $n \gg 1$; the expected number of information packets participating in at least two generations is

$$l + h[1 - \bar{\pi}^{(n-1)}] \approx l + h\left[1 - e^{-l/h}\right] < \min\{g, 2l\}$$

for $n \gg 1$ and $l > 0$.

Claim 14: The probability that two generations overlap is $1 - \binom{N-2h}{l, l, N-2h-2l} / \binom{N-h}{l}^2$. The number of generations overlapping with any one generation G_i is then

$$\mathrm{Binom}\left(n-1, \left[1 - \binom{N-2h}{l, l, N-2h-2l} / \binom{N-h}{l}^2\right]\right).$$

The following Theorem 15 quantifies the expected amount of help a generation may receive from previously decoded generations in terms of common information packets. In the next subsection, we use Corollary 5 and Theorem 15 for a heuristic analysis of the expected throughput performance of the random annex code.

Theorem 15: For any $I \subset \{1, 2, \ldots, n\}$ with $|I| = s$, and any $j \in \{1, 2, \ldots, n\} \backslash I$,

$$\Omega(s) = E[|\,(\cup_{i \in I} G_i) \cap G_j|] = g \cdot [1 - \bar{\pi}^s] + sh \cdot \pi\bar{\pi}^s \quad (19)$$

where $|B|$ denotes the cardinality of set B. When $n \to \infty$, if $\frac{l}{h} \to \alpha$ and $\frac{s}{n} \to \beta$, and let $\omega(\beta) = \Omega(s)$, then $\omega(\beta) \to h\left[(1+\alpha)\left(1 - e^{-\alpha\beta}\right) + \alpha\beta e^{-\alpha\beta}\right]$.

Proof: Please refer to Appendix D. ■

C. Expected Throughput Analysis: The Algorithm

Given the overlapping structure, we next describe an analysis of the expected number of coded packets a receiver needs to collect in order to decode all N information packets of $\mathcal{F}$ when they are encoded by the random annex code. We base our analysis on Theorem 15 above, Corollary 5 in Section IV, and also (3) in Section III, and use the mean value for every quantity involved.

By the time when s ($s = 0, 1, \ldots, n-1$) generations have been decoded, for any one of the remaining $(n-s)$ generations, on the average $\Omega(s)$ of its participating information packets have been decoded, or equivalently, $(g - \Omega(s))$ of them are not yet resolved. If for any one of these remaining generations the receiver has collected enough coded packets to decode its unresolved packets, that generation becomes the $(s+1)$th decoded; otherwise, if no such generation exists, decoding fails.

The quantity $\eta_g(x)$ defined in (3) in Section III estimates the number of coded packets from a generation of size g adequate for collecting x linearly independent equations. By extending the domain of $\eta_g(x)$ from integers to real numbers, we can estimate that the number of coded packets needed for the $(s+1)$th decoded generation should exceed $m'_s = \lceil \eta_g(g - \Omega(s)) \rceil$. Since in the random annex code, all generations are randomly scheduled with equal probability, for successful decoding, we would like to have at least m'_0 coded packets belonging to one of the generations, at least m'_1 belonging to another, and so on. Then Corollary 5 in Section IV can be applied to estimate the total number of coded packets needed to achieve these minimum requirements for the numbers of coded packets.

The algorithm for our heuristic analysis is listed as follows:

1) Compute $\Omega(s-1)$ for $s = 1, \ldots, n$ using Theorem 15;
2) Compute $m'_s = \lceil \eta_g(g - \Omega(s-1)) \rceil$ for $s = 1, 2, \ldots, n$ using (3);
3) Map m'_s ($s = 1, 2, \ldots, n$) into A values m_j ($j = 1, 2, \ldots, A$) so that $m_j = m'_{k_{j-1}+1} = m'_{k_{j-1}+2} = \cdots = m'_{k_j}$, for $j = 1, 2, \ldots, A$, $k_0 = 0$ and $k_A = n$;
4) Evaluate (10) in Corollary 5 with the A, k_js, and m_js obtained in Step 3), as an estimate for the expected number of coded packets needed for successful decoding.

Remark 2: The above Step 3) is viable because $\Omega(s)$ is nondecreasing in s, $\eta_g(x)$ is non-decreasing in x for fixed g, and thus m'_s is non-increasing in s.

Although our analysis is heuristic, we will see in the next section that the estimate closely follows the simulated average performance curve of the random annex coding scheme.

D. Numerical Evaluation and Simulation Results

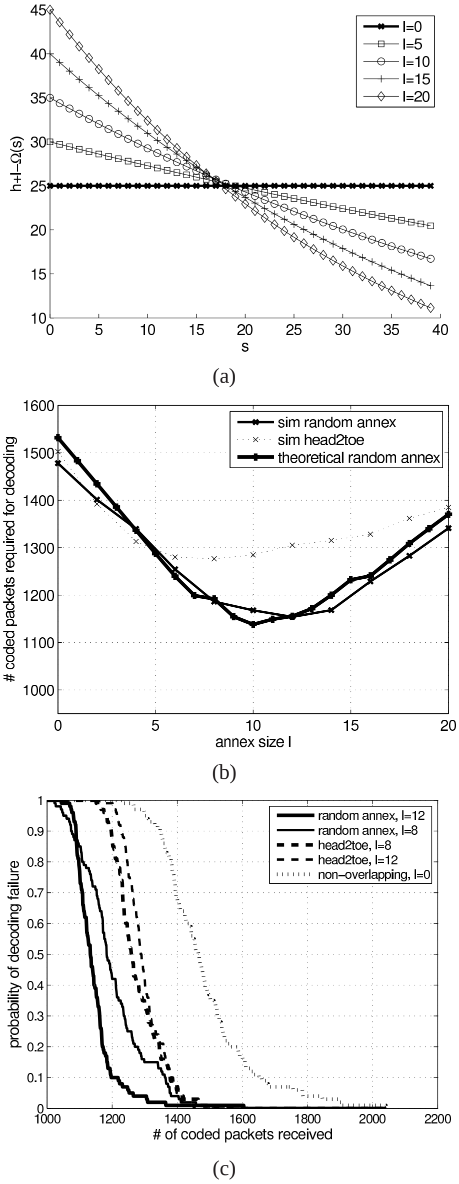
1) Throughput vs. Complexity in Fixed Number of Generations Schemes: Our goal here is to find out how the annex size l affects the decoding latency of the scheme with fixed base generation size h and the total number of information packets N (and consequently, the number of generations n). Note that the generation size $g = h + l$ affects the computational complexity of the scheme, and hence we are actually looking at the tradeoff between throughput and complexity.

Figure 3 shows both the analytical and simulation results when the total number N of information packets is 1000 and the base generation size h is 25. Figure 3(a) shows $h + l - \Omega(s)$ for $s = 0, 1, \dots, n$ with different annex sizes. Recall that $\Omega(s)$ is the expected size of the overlap of the union of s generations with any one of the leftover $n - s$ generations. After the decoding of s generations, for any generation not yet decoded, the expected number of information packets that still need to be resolved is then $h + l - \Omega(s)$. We observe that the $h + l - \Omega(s)$ curves start from $h + l$ for $s = 0$ and gradually descends, ending somewhere above $h - l$, for $s = n - 1$.

Recall that we measure throughput by decoding latency (Section II-F). Figure 3(b) shows the expected performance of the random annex code, along with the performance of the head-to-toe overlapping code and the non-overlapping code ($l = 0$). Figure 3(c) shows the probability of decoding failure of these codes versus the number of coded packets collected.

- Our analysis for the expected decoding latency closely matches the simulation results.
- Figure 3(b) shows that by fixing the file size N and the base generation size h, the expected decoding latency decreases roughly linearly with increasing annex size l, up to $l = 12$ for the random annex scheme and up to $l = 8$ for the head-to-toe scheme. Meanwhile, the decoding cost per information packet is quadratic in $g = h + l$. Beyond the optimal annex size, throughput cannot be further increased by raising computational cost.
- The random annex code outperforms head-to-toe overlapping at their respective optimal points. Both codes outperform the non-overlapping scheme.
- As more coded packets are collected, the probability of decoding failure of the random annex code converges to 0 faster than that of the head-to-toe and that of the non-overlapping scheme.

Overlaps provide a tradeoff between computational complexity and decoding latency.



(a)

(b)

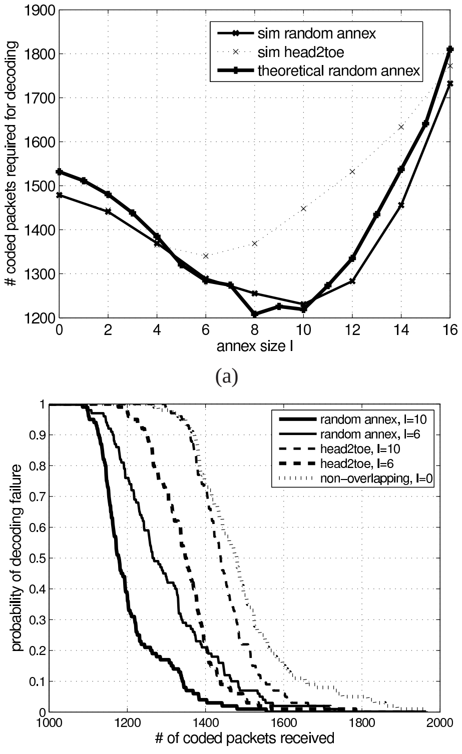
(c)

Fig. 3. $N = 1000$, $h = 25$, $q = 256$: (a) Difference between the generation size and the expected size of overlap with previously decoded generations $(h + l - \Omega(s))$; (b) Expected number of coded packets needed for successful decoding versus annex size l; (c) Probability of decoding failure

2) Enhancing Throughput in Fixed Complexity Schemes: Our goal here is to see if we can choose the annex size to optimize the throughput with negligible sacrifice in complexity. To this end, we fix the extended generation size $g = h + l$ and vary only the annex size l. Consequently, the computational complexity for coding does not increase when l increases. Actually, since some of the information packets in a generation of size g could already be solved while decoding other generations, the remaining information packets in this generation can be solved in a system of linear equations of fewer than g unknowns, and as a result increasing l might decrease the decoding complexity.

Figure 4 shows both the analytical and simulation results for the code performance when the total number N of information packets is fixed at 1000 and size g of extended generation fixed at 25.

- Again our analytical results agree with simulation results very well;
- It is interesting to observe that, without raising computational complexity, increasing annex size properly can still give non-negligible improvement to throughput;
- Figure 4(a) shows a roughly linear improvement of throughput with increasing l, up to $l = 10$ for the random



(a)

(b)

Fig. 4. $N = 1000$, $g = h+l = 25$, $q = 256$: (a) Expected number of coded packets needed for successful decoding versus annex size l; (b) Probability of decoding failure

annex scheme and up to $l = 6$ for the head-to-toe scheme. Increasing l beyond affects throughput adversely;

- The random annex code again outperforms head-to-toe overlapping at their optimal points. Both codes outperform the non-overlapping scheme;
- We again observe that the probability of decoding failure of the random annex code converges faster than those of the head-to-toe and the non-overlapping schemes.

When the overlap size increases, we either have larger generations with unchanged number of generations, or a larger number of generations with unchanged generation size. In both cases the decoding latency would increase if we neglected the effect of overlaps during the decoding process. If we make use of the overlap in decoding, on the other hand, the larger the overlap size, the more help the generations can lend to each other in decoding and, hence, reducing the decoding latency. Two canceling effects result in a non-monotonic relationship between throughput and overlap size.

The effect of generation size on the throughput of random annex codes is further illustrated in Figure 5. Figure 5 plots the optimal expected decoding latency achievable by random annex codes and the corresponding optimal annex size versus the generation size for $N = 1000$ and $q = 16$. The plotted values are calculated using the algorithm listed in Section VI-C.

We can see from Figure 5 that with the random annex code and a generation size of 20, the expected throughput is better than what can be achieved with coding over disjoint generations and a generation size of 50. The reduction in computational complexity is considerable. Capturing the optimal overlap size in terms of other parameters of the code is our object of interest in the future.

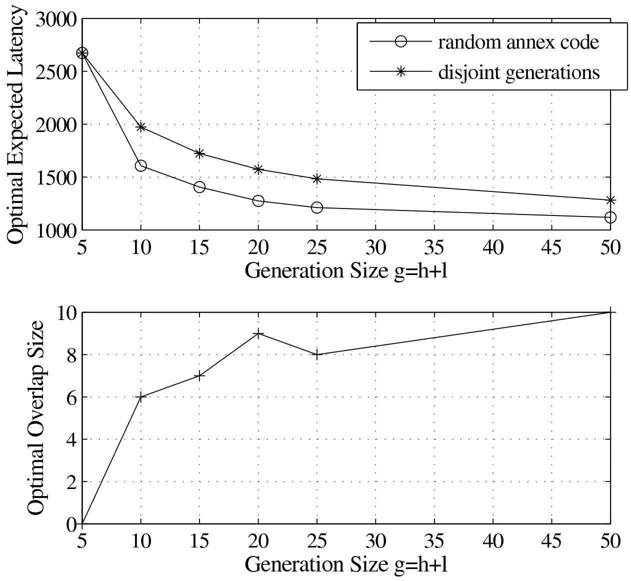


Fig. 5. Optimal expected decoding latency and the optimal overlap size with random annex codes. $N = 1000$, $q = 16$

Appendix A
Proof of Claim 1

For $i = 1, 2, \ldots, n$ and any $s \geq g$, we have

$$\begin{aligned}
&\ln \text{Prob}\big\{M(g,g) \leq s\big\}\\
&\quad= \ln \prod_{k=0}^{h-1}(1-q^{k-s}) = \sum_{k=0}^{g-1} \ln(1-q^{k-s})\\
&\quad= -\sum_{k=0}^{g-1}\sum_{j=1}^{\infty}\frac{1}{j}q^{(k-s)j} = -\sum_{j=1}^{\infty}\frac{1}{j}\sum_{k=0}^{g-1}q^{j(k-s)}\\
&\quad= -\sum_{j=1}^{\infty}\frac{1}{j}q^{-js}\frac{q^{jg}-1}{q^{j}-1}\\
&\quad= -\,q^{-(s-g)}\sum_{j=1}^{\infty}\frac{1}{j}q^{-(j-1)(s-g)}\frac{1-q^{-jg}}{q^{j}-1}\\
&\quad> q^{-(s-g)}\sum_{j=1}^{\infty}\frac{1}{j}\frac{1-q^{-jg}}{1-q^{j}}\\
&\quad= q^{-(s-g)}\ln \text{Prob}\big\{M(g,g) \leq g\big\}\\
&\quad> q^{-(s-g)}\lim_{h\to\infty, q=2}\ln \text{Prob}\big\{M(g,g) \leq g\big\}
\end{aligned}$$

The claim is obtained by setting

$$\alpha_{q,g} = -\ln \text{Prob}\big\{M(g,g) \leq g\big\},$$

and

$$\alpha_{2,\infty} = -\lim_{g\to\infty, q=2}\ln \text{Prob}\big\{M(g,g) \leq g\big\}.$$

APPENDIX B
PROOFS OF GENERALIZED RESULTS OF COLLECTOR'S BROTHERHOOD PROBLEM

Proof of Theorem 2

Our proof generalizes the symbolic method of [10].

Let ξ be the event that the number of copies of coupon G_i is at least m_i for every $i = 1, 2, \ldots, n$. For integer $t \geq 0$, let $\xi(t)$ be the event that ξ has occurred after a total of t samplings, and let $\bar{\xi}(t)$ be the complementary event. Then, the tail probability $\mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) > t] = \mathrm{Prob}[\bar{\xi}(t)] = \nu_t$.

To derive ν_t, we introduce an operator f acting on an n-variable polynomial g. f removes all monomials $x_1^{w_1} x_2^{w_2} \ldots x_n^{w_n}$ in g satisfying $w_1 \geq m_1, \ldots, w_n \geq m_n$. Note that f is a linear operator, i.e., if g_1 and g_2 are two polynomials in the same n variables, and a and b two scalars, we have $af(g_1) + bf(g_2) = f(ag_1 + bg_2)$.

Each monomial in $(x_1 + \cdots + x_n)^t$ corresponds to one of the n^t possible outcomes of t samplings, with the exponent of x_i being the number of copies of coupon G_i. Since the samplings are independent, the probability of an outcome $x_1^{w_1} x_2^{w_2} \ldots x_n^{w_n}$ is $\rho_1^{w_1} \rho_2^{w_2} \ldots \rho_n^{w_n}$. Hence, the probability of $\bar{\xi}(t)$ is $f((x_1 + \cdots + x_n)^t)$, when evaluated at $x_i = \rho_i$ for $i = 1, 2, \ldots n$, i.e.,

$$\nu_t = f((x_1 + \cdots + x_n)^t)|_{x_i = \rho_i, i=1,\ldots,n}. \tag{20}$$

Hence, (20) and (7) lead to

$$\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z) = \sum_{t \geq 0} f\left((x_1 + \cdots + x_n)^t\right) z^t|_{x_i = \rho_i, i=1,\ldots,n}.$$

The identity

$$\int_0^\infty \frac{1}{t!} y^t e^{-y} dy = 1$$

and the linearity of the operator f imply that

$$\begin{aligned}
\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z) &= \int_0^\infty \sum_{t \geq 0} \frac{f\left((x_1 + \cdots + x_n)^t\right)}{t!} z^t y^t e^{-y} dy \\
&= \int_0^\infty f\Big(\sum_{t \geq 0} \frac{(x_1 zy + \cdots + x_n zy)^t}{t!}\Big) e^{-y} dy \\
&= \int_0^\infty f\left(\exp(x_1 zy + \cdots + x_n zy)\right) e^{-y} dy \quad (21)
\end{aligned}$$

evaluated at $x_i = \rho_i, i = 1, \ldots, n$.

We next find the sum of the monomials in the polynomial expansion of $\exp(x_1 + \cdots + x_n)$ that should be removed under f. Clearly, this sum should be $\prod_{i=1}^n \left(e^{x_i} - S_{m_i}(x_i)\right)$, where S is defined in (5) and (6)). Therefore,

$$\begin{aligned}
&f\left(\exp(x_1 zy + \cdots + x_n zy)\right)|_{x_i = \rho_i, i=1,\ldots,n} \\
&= e^{zy} - \prod_{i=1}^n \left(e^{\rho_i zy} - S_{m_i}(\rho_i zy)\right).
\end{aligned}$$

$$\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z) = \int_0^\infty \left[e^{zy} - \prod_{i=1}^n \left(e^{\rho_i zy} - S_{m_i}(\rho_i zy)\right)\right] e^{-y} dy \tag{22}$$

Proof of Corollary 3

Note that

$$\begin{aligned}
\varphi_{T(\boldsymbol{\rho},\mathbf{m})}(z) &= \sum_{t=0}^\infty \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) > t] z^t \\
&= \sum_{t=0}^\infty \sum_{j=t+1}^\infty \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) = j] z^t \\
&= \sum_{j=1}^\infty \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) = j] \sum_{t=0}^{j-1} z^t
\end{aligned}$$

$$E[T(\boldsymbol{\rho}, \mathbf{m})] = \sum_{j=1}^\infty j \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) = j] = \varphi_{T(\boldsymbol{\rho},\mathbf{m})}(1).$$

Similarly,

$$\begin{aligned}
\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(z) &= \sum_{t=0}^\infty t \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) > t] z^{t-1} \\
&= \sum_{j=1}^\infty \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) = j] \sum_{t=0}^{j-1} t z^{t-1} \\
\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(1) &= \sum_{j=1}^\infty \frac{1}{2} j(j-1) \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) = j].
\end{aligned}$$

Hence,

$$\begin{aligned}
E[T(\boldsymbol{\rho}, \mathbf{m})^2] &= \sum_{j=1}^\infty j^2 \mathrm{Prob}[T(\boldsymbol{\rho}, \mathbf{m}) = j] \\
&= 2\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(1) + \varphi_{T(\boldsymbol{\rho},\mathbf{m})}(1),
\end{aligned}$$

and consequently,

$$\mathrm{Var}[T(\boldsymbol{\rho}, \mathbf{m})] = 2\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(1) + \varphi_{T(\boldsymbol{\rho},\mathbf{m})}(1) - \varphi^2_{T(\boldsymbol{\rho},\mathbf{m})}(1).$$

We have

$$\begin{aligned}
&\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(z) = \\
&\int_0^\infty x\Big(e^{-x(1-z)} - \sum_{i=1}^n \rho_i \frac{e^{-\rho_i x(1-z)} - S_{m_i-1}(\rho_i xz) e^{-\rho_i x}}{e^{-\rho_i x(1-z)} - S_{m_i}(\rho_i xz) e^{-\rho_i x}} \cdot \\
&\cdot \prod_{j=1}^n \Big(e^{-\rho_j x(1-z)} - S_{m_j}(\rho_j xz) e^{-\rho_j x}\Big)\Big) dx,
\end{aligned}$$

and from there, we can get $\varphi'_{T(\boldsymbol{\rho},\mathbf{m})}(1)$ and $\mathrm{Var}[T(\boldsymbol{\rho}, \mathbf{m})]$.

Proof of Theorem 4

We again apply the Newman-Shepp symbolic method. Similar to the proof of Theorem 2, we introduce an operator f acting on an n-variable polynomial g. For a monomial $x_1^{w_1} \ldots x_n^{w_n}$, let i_j be the number of exponents w_u among $w_1, \ldots, w_n$ satisfying $w_u \geq k_j$, for $j = 1, \ldots, A$. f removes all monomials $x_1^{w_1} \ldots x_n^{w_n}$ in g satisfying $i_1 \geq k_1, \ldots, i_A \geq k_A$ and $i_1 \leq \cdots \leq i_A$. f is again a linear operator. One can see that

$$\begin{aligned}
&\varphi_{U(\mathbf{m},\mathbf{k})}(z) = \qquad (23) \\
&\int_0^\infty f\left(\exp(x_1 zy + \cdots + x_n zy)\right) e^{-y} dy|_{x_1 = x_2 = \cdots = x_n = \frac{1}{n}}.
\end{aligned}$$

We choose integers $0 = i_0 \leq i_1 \leq \cdots \leq i_A \leq i_{A+1} = n$, such that $i_j \geq k_j$ for $j = 1, \ldots, A$, and then partition indices $\{1, \ldots, n\}$ into $(A+1)$ subsets $\mathcal{I}_1, \ldots, \mathcal{I}_{A+1}$, where $\mathcal{I}_j (j = 1, \ldots, A+1)$ has $i_j - i_{j-1}$ elements. Then

$$\prod_{j=1}^{A+1} \prod_{i \in \mathcal{I}_j} (S_{m_{j-1}}(x_i) - S_{m_j}(x_i)) \tag{24}$$

equals the sum of all monomials in $\exp(x_1 + \cdots + x_n)$ with $(i_j - i_{j-1})$ of the n exponents smaller than m_{j-1} but greater than or equal to m_j, for $j = 1, \ldots, A+1$. (Here S is as defined by (5-6).) The number of such partitions of $\{1, \ldots, n\}$ is equal to $\binom{n}{n-i_A, \ldots, i_2 - i_1, i_1} = \prod_{j=0}^{A} \binom{i_{j+1}}{i_j}$. Finally, we need to sum the terms of the form (24) over all partitions of all choices of $i_1, \ldots, i_A$ satisfying $k_j \leq i_j \leq i_{j+1}$ for $j = 1, \ldots, A$:

$$\begin{aligned} &f\left(\exp(x_1 zy + \cdots + x_n zy)\right)|_{x_1 = \cdots = x_n = \frac{1}{n}} = \exp(zy) - \\ &\sum_{\substack{(i_0, i_1, \ldots, i_{A+1}): \\ i_0 = 0, i_{A+1} = n \\ i_j \in [k_j, i_{j+1}] \\ j = 1, 2, \ldots, A}} \prod_{j=0}^{A} \binom{i_{j+1}}{i_j} \left[S_{m_j}(\frac{zy}{n}) - S_{m_{j+1}}(\frac{zy}{n}) \right]^{i_{j+1} - i_j}. \end{aligned} \tag{25}$$

Bringing (25) into (23) gives our result in Theorem 4.

Appendix C
Proof of Theorem 8

$$\begin{aligned} &E[W(\boldsymbol{\rho}, \mathbf{g})] \\ &= \sum_{\mathbf{m}} \left(\prod_{i=1}^{n} \Pr[M_i = m_i] \right) E[T(\boldsymbol{\rho}, \mathbf{m})] \\ &= \int_0^\infty \left[1 - \prod_{i=1}^{n} \sum_{m_i} \Pr[M_i = m_i](1 - S_{m_i}(\rho_i x) e^{-\rho_i x}) \right] dx \qquad (26) \\ &= \int_0^\infty \left(1 - \prod_{i=1}^{n} \left(1 - e^{-\rho_i x} E_{M_i}\left[S_{M_i}(\rho_i x) \right] \right) \right) dx. \end{aligned}$$

(26) comes from the distributivity.

Since

$$E_{M_i}\left[S_{M_i}(\rho_i x) \right] = \sum_{j=0}^{\infty} \frac{(\rho_i x)^j}{j!} \Pr[M_i > j],$$

by Claim 1,

$$\begin{aligned} &E_{M_i}\left[S_{M_i}(\rho_i x) \right] \\ &< S_{g_i}(\rho_i x) + \sum_{j=g_i}^{\infty} \frac{(\rho_i x)^j}{j!} \alpha_{q,g} q^{-(j-g)} \\ &= S_{g_i}(\rho_i x) + \alpha_{q,g_i} q^{g_i} e^{\rho_i x / q} - \alpha_{q,g_i} q^{g_i} S_{g_i}(\rho_i x / q), \end{aligned}$$

where

$$\alpha_{q,g_i} = -\ln \Pr\left\{ M(g_i, g_i) \leq g_i \right\} = -\sum_{k=0}^{g_i - 1} \ln\left(1 - q^{k - g_i} \right)$$

for $i = 1, 2, \ldots, n$.

Hence, we have (17).

Expression (18) for $E[W^2(\boldsymbol{\rho}, \mathbf{g})]$ can be derived in the same manner, and then the expression for Var$[W(\boldsymbol{\rho}, \mathbf{g})]$ immediately follows.

Appendix D
Proof of Theorem 15

Without loss of generality, let $I = \{1, 2, \ldots, s\}$ and $j = s+1$, and define $\mathcal{R}_s = \cup_{i=1}^{s} R_i$, $\mathcal{B}_s = \cup_{i=1}^{s} B_i$, and $\mathcal{G}_s = \cup_{i=1}^{s} G_i$ for $s = 0, 1, \ldots, n-1$. Then, $E\left[|(\cup_{i \in I} G_i) \cap G_j| \right] = E\left[|\mathcal{G}_s \cap G_{s+1}| \right]$. For any two sets X and Y, we use $X + Y$ to denote $X \cup Y$ when $X \cap Y = \emptyset$.

$$\begin{aligned} \mathcal{G}_s \cap G_{s+1} =& (\mathcal{B}_s + \mathcal{R}_s \backslash \mathcal{B}_s) \cap (B_{s+1} + R_{s+1}) \\ =& \mathcal{B}_s \cap R_{s+1} + \mathcal{R}_s \cap B_{s+1} + (\mathcal{R}_s \backslash \mathcal{B}_s) \cap R_{s+1}, \end{aligned}$$

and therefore

$$\begin{aligned} E[|\mathcal{G}_s \cap G_{s+1}|] =& E[|\mathcal{B}_s \cap R_{s+1}|] + \\ & E[|\mathcal{R}_s \cap B_{s+1}|] + E[|(\mathcal{R}_s \backslash \mathcal{B}_s) \cap R_{s+1}|]. \end{aligned} \tag{27}$$

Using Claim 11, we have

$$E[|\mathcal{B}_s \cap R_{s+1}|] = sh\pi, \tag{28}$$

$$E[|\mathcal{R}_s \cap B_{s+1}|] = h[1 - (1 - \pi)^s], \tag{29}$$

$$E[|(\mathcal{R}_s \backslash \mathcal{B}_s) \cap R_{s+1}|] = (n - s - 1) h\pi[1 - (1 - \pi)^s], \tag{30}$$

where π is as defined in Claim 11. Bringing (28)-(30) into (27), we obtain (19).

Furthermore, when $n \to \infty$, if $l/h \to \alpha$ and $s/n \to \beta$, then

$$\begin{aligned} E[|\mathcal{G}_s \cap G_{s+1}|] =& g \cdot [1 - \bar{\pi}^s] + sh \cdot \pi \bar{\pi}^s \\ \to& h(1 + \alpha) \Big[1 - \Big(1 - \frac{\alpha}{n-1} \Big)^{n\beta} \Big] + \\ & h\alpha\beta \Big(1 - \frac{\alpha}{n-1} \Big)^{n\beta} \\ \to& h \Big[(1 + \alpha)(1 - e^{-\alpha\beta}) + \alpha\beta e^{-\alpha\beta} \Big] \\ =& h \Big[1 + \alpha - (1 + \alpha - \alpha\beta) e^{-\alpha\beta} \Big] \end{aligned}$$

Acknowledgement

The authors would like to thank the anonymous reviewers for helpful suggestions to improve the presentation of the paper.

References

[1] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE International Symposium on Information Theory (ISIT'03)*, 2003, p. 442.

[2] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, vol. 4, Miami, FL, Mar. 2005, pp. 2235–2245.

[3] Z. Liu, C. Wu, B. Li, and S. Zhao, "UUSee: Large-scale operational on-demand streaming with random network coding," in *Proc. the 30th IEEE Conference on Computer Communications (INFOCOM'10)*, San Diego, California, Mar. 2010.

[4] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

[5] P. Maymounkov, N. Harvey, and D. S. Lun, "Methods for efficient network coding," in *Proc. 44th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 2006.

[6] D. Silva, W. Zeng, and F. Kschischang, “Sparse network coding with overlapping classes,” in *Proc. Workshop on Network Coding, Theory, and Applications (NetCod ’09)*, Lausanne, Switzerland, Jun. 2009, pp. 74–79.

[7] A. Heidarzadeh and A. Banihashemi, “Overlapped chunked network coding,” in *IEEE Information Theory Workshop (ITW’10)*, Jan. 2010, pp. 1–5.

[8] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. New York: John Wiley & Sons Inc., 1968, vol. 1, pp. 224–225.

[9] C. Fragouli and E. Soljanin, *Network Coding Applications*, ser. Foundations and Trends in Networking. Hanover, MA: now Publishers Inc., Jan. 2008, vol. 2, no. 2, pp. 146–147.

[10] D. Newman and L. Shepp, “The double dixie cup problem,” *The American Mathematical Monthly*, vol. 67, no. 1, pp. 58–61, Jan. 1960.

[11] D. Foata and D. Zeilberger, “The collector’s brotherhood problem using the Newman-Shepp symbolic method,” *Algebra Universalis*, vol. 49, no. 4, pp. 387–395, 2003.

[12] A. Boneh and M. Hofri, “The coupon-collector problem revisited – a survey of engineering problems and computational methods,” *Stochastic Models*, vol. 13, pp. 39 – 66, 1997. [Online]. Available: http://www.informaworld.com/10.1080/15326349708807412

[13] P. Flajolet, D. Gardy, and L. Thimonier, “Birthday paradox, coupon collectors, caching algorithms and self-organizing search,” *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.

[14] L. Flatto, “Limit theorems for some random variables associated with urn models,” *The Annals of Probability*, vol. 10, no. 4, pp. 927–934, 1982. [Online]. Available: http://www.jstor.org/stable/2243548

[15] A. N. Myers and H. S. Wilf, “Some new aspects of the coupon collector’s problem,” *SIAM Rev.*, vol. 48, no. 3, pp. 549–565, 2006.

[16] P. Erdös and A. Rényi, “On a classical problem of probability theory,” *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, vol. 6, pp. 215–220, 1961.