ARQ with Doped Fountain Decoding

Silvija Kokalj-Filipović, Predrag Spasojević, Emina Soljanin, and Roy Yates

***Abstract*—We analyze the benefits of an automatic repeat-request (ARQ) scheme employed in doped belief-propagation decoding for increasing the throughput of fountain encoded data transmissions. The proposed doping mechanism selects doping symbols randomly from the set of input symbols contributing to degree-two output symbols. Here an output symbol is the encoded symbol whose code-graph links to input symbols decoded thus far have been severed. This doping approach always ensures releasing of at least one output symbol, thus increasing the number of degree-one output symbols (the ripple). Using a random walk analysis, we study the belief propagation decoding with degree-two random doping for a fountain code with symbols drawn from an Ideal Soliton distribution. We show that the decoding process is a renewal process whereas the process starts all over afresh after each doping. The approximate *interdoping process* analysis revolves around a random walk model for the ripple size. We model the sequence of the ripple size increments (due to doping and/or decoding) as an iid sequence of shifted and truncated Poisson random variables. This model furnishes a prediction on the number of required doping symbols and, furthermore, the ARQ throughput cost analysis. We also find that the Ideal Soliton significantly outperforms the Robust Soliton distribution in our ARQ-doping scheme.**

I. Introduction

The classical feedback protocols in most communication systems are the automatic repeat request (ARQ) schemes. Here, the receiving end examines packets and requests a retransmission if an error is detected. The more efficient incremental redundancy (IR) hybrid (H)ARQ protocol operates as follows. At the transmitter, the information is encoded by a ”mother” code. Initially, only a subset of codeword symbols is transmitted. If the decoding is not successful, the transmitter is requested to send additional (pre-determined) codeword symbols through a 1-bit feedback. The second decoding attempt is based on the new symbols combined with those previously received, and so on. The standard measure of ARQ protocol efficiency is throughput χ, defined as the average number of information bits decoded successfully at the receiver per transmitted encoded bit. Here we consider a communication scheme where k symbols/packets are encoded using a Fountain-type rateless code and a ($\log k$)-bit feedback requests transmission of those symbols/packets which will restart the stalled message-passing decoding process. A fountain code combines a random number (*degree*) of input symbols into an encoded symbol independently for each symbol and based on a given degree distribution $\Omega(d)$. Hence, this feedback protocol can be viewed as an IR-HARQ scheme where the additional degree-one code symbol sent by the transmitter is determined by the decoding process. An analysis of an IR-HARQ scheme based on Raptor rateless codes [8] and maximum-likelihood decoding is given in [9].

This scheme can also be viewed as a doping scheme, where the stalled message passing decoding process is doped with that input symbol which will enable its continuation. Doping for improved decoding was first described in [10] as a technique that enables iterative decoding of serially concatenated codes. [1] reports a two-stage scheme where a code is sent in the first stage, while in the second stage the encoder maintains a dialog with belief-propagation (*BP*) decoder, enhanced by a doping algorithm. In [4], we provide numerical results which illustrate how doped BP decoding of Fountain codes can be employed to reduce the delay of data collection in a wireless sensor network.

In this work we analyze the benefits of doped BP decoding for increasing the throughput of Fountain encoded data. In our analysis, we use Fountain codes based on an Ideal Soliton distribution of code symbol degrees, and also illustrate the benefits of doped decoding for Robust Soliton based Fountain codes, such as LT codes, using simulations. The proposed doping mechanism selects doping symbols randomly from the set of input symbols contributing to degree-two output symbols. Here an output symbol is the encoded symbol whose code graph links to all decoded input symbols have been severed. This doping approach always ensures releasing of at least one output symbol, thus increasing the number of degree-one output symbols (the ripple). We refer to the set of such input symbols as the *releasing set*. Simulation results show that random selection of a symbol from the releasing set performs equally well as the greedy selection where the symbol present in the largest number of degree-two output symbols is selected. This fact simplifies our random-walk *(RW)* representation of the ripple evolution process. The RW-based model represents the key contribution of this paper. It furnishes the final cost analysis with a prediction on the number of required doping symbols.

II. System Model and Problem Formulation

We consider a data source that produces k unique input symbols, where each symbol is a binary vector of length q, also referred to as packet. Let the encoder linearly combine random subsets of input symbols into coded symbols of a given degree distribution $\omega(d)$. Hence, a random code from the ensemble of Fountain codes with degree distribution $\omega(d)$ and its generating polynomial $\Omega(x) = \sum_{d=1}^{k} \Omega_d x^d$, where $\Omega_d = \omega(d)$, is produced. We assume a BP decoder that starts decoding once it collects $s \geq k$ coded symbols. We allow the decoder to request an input symbol from the source at any time the decoding process stalls. The cost of requesting the input symbol is $\log k$. We refer to such a decoding process as belief propagation with ARQ. We refer to the requested input symbols as *doped symbols*. Let the expected number

S. Kokalj-Filipović, P. Spasojević and Roy Yates are with Wireless Information Network Laboratory (WINLAB), Department of Electrical and Computer Engineering, Rutgers University, North Brunswick, NJ 08902, USA (e-mail: {skokalj,spasojev,ryates}@winlab.rutgers.edu).

E. Soljanin is with Lucent-Alcatel, Bell Labs, Murray Hill, NJ, USA (email: emina@research.bell-labs.com).

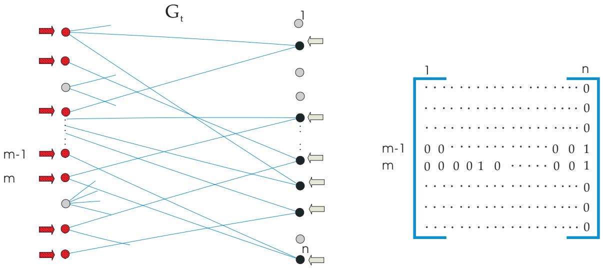

Fig. 1. In the graph $\mathbf{G_t}$, representing the stalled decoding process at time t, we identify nodes on the left side (input symbols corresponding to rows of the incidence matrix) connected to right-hand-side nodes of degree two (output nodes corresponding to columns of weight two, represented by black nodes, and pointed to by black arrows), and then uniformly at random select one such input symbol to unlock the decoder. The set of symbols we are selecting from is represented by red nodes, indicated by red arrows.

of doped symbols needed to complete the decoding of k symbols be s_d. We are interested in the total cost of decoding $s_t = sq + s_d\,(q + \log k)$ (in bits). We compare it with the expected cost for decoding of LT codes, expressed in number of coded symbols needed to decode k input symbols up to a small error constant δ, which is of order $O(k + \sqrt{k}\log^2 k/\delta)$. The expected number of doped symbols s_d is affected by the choice of $\Omega(x)$ and by the criterion for selecting doped input symbols. Choosing a good distribution $\Omega(x)$ is not easy, as it needs to satisfy many contradicting requirements. High-degree code symbols are good for decreasing the probability of uncovered packets. However, other requirements are more important for proper behavior of the belief propagation decoding process, especially the right amount of degree one and degree two code symbols. We limit the set of candidate degree distributions $\Omega(x)$ to Robust Soliton, being the best known practical solution for Fountain codes, as argued in [5], and Ideal Soliton distribution, the best theoretical (asymptotic) solution [5], [7]. It is well known that Ideal Soliton's expected behavior is perfect for Fountain codes decoded by an iterative decoder, but the large variance may cause the absence of degree-one symbols (*the ripple*) in the collected sample of code symbols, thus stalling the BP process. This is the reason why Robust Soliton is used as a choice degree distribution for rateless erasure codes. For Robust Soliton, the probability of one-degree symbols is overdesigned in order to prevent stalling. However, redistribution of the probability mass from higher degrees to degree-one increases the likelihood of uncovered packets.

With respect to doping criterion, we introduce a strategy dubbed *degree-2 doping*. Let us denote the graph describing the decoding (BP) process at time t by $\mathbf{G_t}$. To unlock the belief propagation process stalled at time (iteration) t, the degree-2 doping strategy selects the doping symbol from the set of input symbols connected to degree-two output symbols in graph $\mathbf{G_t}$, as illustrated in Figure 1.

III. Doped Belief Propagation Decoding

According to our model, code symbols are linear combinations of k unique input symbols, indexed by the set $\{1,\cdots,k\}$. We start with a decoding matrix $\mathbf{S_0} = [s_{ij}]_{k\times k}$, where code symbols are represented by the columns, so that $s_{ij} = 1$ iff the jth code symbol contains the ith input symbol. Number of ones in the column corresponds to the degree of the associated code symbol. Input symbols covered by the code symbols with degree one constitute the ripple. In the first step of the decoding process, one input symbol in the ripple is processed by being removed from all code symbols which have it as a neighbor in the associated graph $\mathbf{G_0}$. If the index of the input symbol is m, this effectively removes the mth row of the matrix, thus creating the new decoding matrix $\mathbf{S_1} = [s_{ij}]_{(k-1)\times k}$. We refer to the code symbols modified by the removal of the processed input symbol as output symbols. Output symbols of degree one may cover new input symbols and thus modify the ripple. Hence, the distribution of output symbol degrees changes to $\Omega_1(x)$. At each subsequent step of the decoding process one input symbol in the ripple is processed by being removed from all output symbols which have it as a neighbor and all such output symbols that subsequently have exactly one remaining neighbor are released to cover their remaining neighbor. Consequently, the support of the output symbol degrees after ℓ input symbols have been processed is $d \in \{1,\cdots,k-\ell\}$, and the resulting output degree distribution is denoted by $\Omega_\ell(x)$.

Now, let us assume that input symbols to be processed are not taken from the ripple, but instead provided to the decoder as side information. We refer to this mechanism of processing input symbols obtained as side information as *doping*. We model the ℓth step of the decoding/doping process by selecting a row uniformly at random from the set of $(k-\ell)$ rows in the current decoding matrix $\mathbf{S}_\ell = [s_{ij}]_{(k-\ell)\times k}$, and removing it from the matrix. Thus the degrees of the output symbols are changing in the same manner as with the regular decoding process, except for the ripple size, which does not decrease by one with each processed input symbol.

We now present the model of such a doping (decoding) process through the column degree distribution at each decoding/doping round. After ℓ rounds or, equivalently, when there are $k-\ell$ rows in the decoding matrix, the number of ones in a column is denoted by $A_{k-\ell}$. The probability that the column is of degree d, when its length is $k-\ell-1,\ \ell\in\{1,\cdots,k-2\}$, is defined iteratively

$$P\left(A_{k-\ell-1}=d\right)=$$
$$=P\left(A_{k-\ell}=d\right)\left(1-\frac{d}{k-\ell}\right)+P\left(A_{k-\ell}=d+1\right)\frac{d+1}{k-\ell} \tag{1}$$

for $0\le d\le k-\ell$, and $P\left(A_{k-\ell-1}=k-\ell\right)=0$.

Let the starting distribution of the column degrees (for the decoding matrix $\mathbf{S_0} = [s_{ij}]_{k\times k}$) be Ideal Soliton, denoted by $\rho(d)$,

$$\rho(d)=\begin{cases}\frac{1}{k}, & d=1,\\ \frac{1}{d(d-1)} & d=2,\cdots,k.\end{cases} \tag{2}$$

By construction, for $l = 0$, $P\left(A_k = d\right) = \rho(d)$, which, together with (1), completely defines the dynamics of the doping process when the Fountain code is based on Ideal Soliton. After rearanging and canceling appropriate terms, for

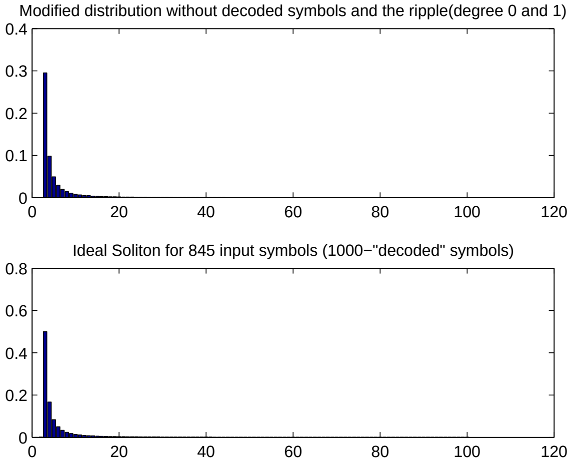

Fig. 2. Density Evolution of Ideal Soliton distribution due to uniform doping. First graph is the distribution of the output symbols after m decodings, for $k = 1000$, second is the Ideal Soliton with support set $\{1, \cdots, (1000 - m)\}$ as if we are starting with the matrix of the same size as the doped matrix.

the random doping case we obtain

$$P\left(A_{k-\ell} = d\right) = \begin{cases} \frac{k-\ell}{k}\left(\mathbf{H_k} - \mathbf{H_{k-\ell-1}}\right) & d = 1, \\ \frac{k-\ell}{k}\rho(d) & d = 2, \cdots, k-\ell, \\ 0 & d > k-\ell, \end{cases} \tag{3}$$

where $\mathbf{H_k}$ is the kth harmonic number, and $P\left(A_{k-\ell} = 0\right) = 1 - \sum_{d \neq 0} P\left(A_{k-\ell} = d\right) \approx \frac{\ell}{k} - \frac{k-\ell}{k}\log\frac{k}{k-\ell}$.

Note that there are approximately as many released coded symbols as there are decoded input symbols. As the ratio of unreleased symbols can be approximated with $\frac{k-\ell}{k}$, the probability distribution $\omega_\ell^u(d)$ of the unreleased output node degrees at any time ℓ remains Ideal Soliton. For classical BP decoding case, the evolution is the same for unreleased output symbols. Hence,

$$\omega_\ell^u(d) = \frac{k}{k-\ell}P\left(A_{k-\ell} = d\right) = \rho(d) \quad \text{for } d = 2, \cdots, k-\ell. \tag{4}$$

A. Random Walk Model

There exist comprehensive and thorough analytical models for the ripple evolution, characterizing the decoding of LT codes [2], [3]. However, the comprehensive nature of those models makes them very complex. For describing the dynamics of a doped decoder, it suffices to consider a simpler model, which attempts to capture the evolution of the ripple for Ideal Soliton in terms of the expected number of decoded symbols before the ripple becomes empty. Figure 2 illustrates how the Ideal Soliton distribution maintains its shape with decoding/doping. This fact is our main motivator for selecting Ideal Soliton Fountain codes for our ARQ-doping scheme. For simplicity, our model ignores the influence of the ripple size to the decoding process. In particular, we ignore the fact that the larger the ripple size is, the higher is the probability that a released symbol is not innovative, and hence does not increase the ripple size. Our model aims to provide a bound on the number of symbols decoded between two dopings, herein referred to as *interdoping yield.* With respect to interdoping yield, we treat the ARQ-enhanced BP process as a renewal process, where the process starts all over after each doping. The time at which the ith doping occurs (equivalently, decoding stalls for the ith time) is a random variable T_i, and the interdoping yield $T_i - T_{i-1}$ is an IID random variable Y. Our goal is to obtain the expected value of Y by modeling the ripple evolution within this renewal process.

We assume that the number of collected encoded symbols is $s = k$. Let us now assume that at time ℓ the number of decoded symbols equals ℓ, and the number of (unreleased) output symbols is $n = k - \ell$, if the expected size of the ripple is much smaller than the number of unreleased symbols. The next decoding iteration processes a random symbol of degree-one from the ripple. As the encoded symbols are made by independently combining random input symbols, we can assume that the input symbol covered by this degree-one symbol is selected uniformly at random from the set of undecoded symbols (again, we ignore the fact that not all ripple symbols are unique). According to the presented matrix-evolution model of the decoding process, this symbol defines the row in the decoding matrix $\mathbf{S}_\ell$ in which the decoder looks for non-zero elements contributing to degree-two columns, in order to identify the released output symbols. Recall that the unreleased output symbol degree distribution polynomial at time ℓ is $\Omega_\ell^u(x) = \sum \Omega_{d,\ell}x^d$, where $d = 2, \cdots, k-\ell$, and $\Omega_{d,\ell} = \omega_\ell^u(d)$. Releasing output symbols by processing a ripple symbol is equivalent to performing $n_2 = n\Omega_{2,\ell}$ independent Bernoulli experiments with probability of success $p_2 = 2/n$, assuming that the ripple size is much smaller than the number of output symbols.

Hence, the number of released symbols is modeled by a discrete random variable Δ with Binomial distribution $\mathbf{B}\left(n\Omega_{2,\ell}, 2/n\right)$, which for large n can be approximated with a Poisson distribution of intensity $2\Omega_{2,\ell}$

$$\begin{aligned} \Pr\left\{\Delta = r\right\} &= \binom{n_2}{r}(p_2)^r\left(1-p_2\right)^{n_2-r} \\ &\geq \frac{(n_2)^r}{r!}(p_2)^r\left(1-p_2\right)^{n_2-r} \\ &\approx \frac{(2\Omega_{2,\ell})^r}{r!}e^{-2\Omega_{2,\ell}}, r = 0, \cdots, n_2, \end{aligned} \tag{5}$$

where we first applied the Stirling approximation to the Binomial coefficient and, also, assumed that n_2 is much larger than r.

According to (4), the number of degree-two output symbols for Ideal Soliton based Fountain code is expected to be $n_2 = \rho(2)n = n/2$, for any decoding iteration ℓ. Hence, the number of released symbols at any decoding step is modeled by random variable Δ_ℓ, with a (truncated) Poisson distribution of intensity one ($\Delta_\ell \sim \wp(1) \equiv \eta(r), r = 0, 1, \cdots$)

$$\begin{aligned} \eta(r) = \Pr\left\{\Delta_\ell = r\right\} &= \frac{\left[2\rho(2)\right]^r}{r!}e^{-2\rho(2)} \\ &= \frac{e^{-1}}{r!}, r = 0, \cdots, n/2. \end{aligned} \tag{6}$$

For each decoding iteration, one symbol is taken from the ripple and Δ_ℓ symbols are added, so that the increments of the ripple process can be described by IID random variables X_ℓ with the probability distribution described by the generating polynomial $I(x) = \sum \eta(d) x^{d-1}$, where $d = 0, \cdots, n/2$.

Let us now describe a doping iteration. If the number of (unreleased) output symbols at time $T_i = t_i$ is n, the number of degree-two output symbols is again expected to be $n_2 = \rho(2)n = n/2$. Degree-two doping selects uniformly at random a row in the decoding matrix $\mathbf{S_{t,i}}$ that has one or more non-zero elements in columns of degree two. The number of released output symbols is equal to the number of non-zero elements in the selected row, contributing to degree-two columns. As the encoded symbols are made by independently combining random input symbols, this is equivalent to randomly selecting a column of degree two to be released, and restarting the ripple with any of its two input symbols. This leaves us with the decoding matrix that has its number of degree-two columns $n_2 - 1$. However, since n_2 is assumed to be large, this can be ignored. Hence, the doping ripple increment $X^D_{t_i}$ is a random variable described by $I^D(x) = \sum \eta(d) x^{d+1}$, which is equivalent to distribution $I(x)$, shifted by two to the right. The ripple size at time $t \in [t_{i-1},\ t_i]$ is described with $S_{t,i} + 2$ where $S_{t,i} = \sum_{j=t_{i-1}}^{t} X_j$ is a partial sum of IID random variables X_j, with probability distribution $I(x)$. Hence, $S_{t,i}$ is a zero mean random walk.

The expected interdoping yield is quantified by the expected time it takes for the ripple random walk $S_{t,i} + 2$ to become zero. Using random walk terminology, we are interested in the statistics of the random-walk stopping time. The stopping time is the time at which the decoding process stalls (counting from the previous doping, where the first decoding round corresponds to the 0th doping which occurred at $T_0 = 0$). Hence, the i-th stopping time (doping) T_i is defined as

$$T_i = \min \{ t_i : S_{t,i} + 2 \leq 0 \} . \tag{7}$$

The random walk $S_{t,i}$ does not lend itself nicely to a stopping-time analysis based on the Wald Identity, having the zero mean and a non-symmetric threshold at zero. Hence, we consider its Markov Chain model. Here, each possible value of the random walk represents a state of the Markov Chain *(MC)* described by the probability transition matrix $\mathbf{P}$. State $v, v \in \{1, \cdots, \lceil n/2 \rceil\}$ corresponds to the ripple of size $v - 1$. The state 1 is the trapping state, with the transition probability $\mathbf{P_{11}} = 1$ modeling the stopped random walk. Hence, based on (6), we have state transition probabilities

$$\begin{aligned} \mathbf{P_{v(v+b)}} &= \eta(1+b), \\ &\quad \text{for } v = 2, \cdots, \lceil n/2 \rceil,\ b = -1, \cdots, \lceil n/2 \rceil - v && (8) \\ \mathbf{P_{11}} &= 1 && (9) \\ \mathbf{P_{vw}} &= 0, \quad \text{otherwise}, && (10) \end{aligned}$$

resulting in a transition probability matrix of the following form

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \eta(0) & \eta(1) & \eta(2) & \cdots & \eta(m-1) \\ 0 & \eta(0) & \eta(1) & \cdots & \eta(m-2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \eta(0) & \eta(1) \end{bmatrix}_{n_2 \times n_2} . \tag{11}$$

The start of the decoding process is modeled by the MC being in the initial state $v = 3$ (equivalent to the ripple of size two). Based on that, the probability of transitioning into the trapping state, while in the step t, is modeled by the following equation

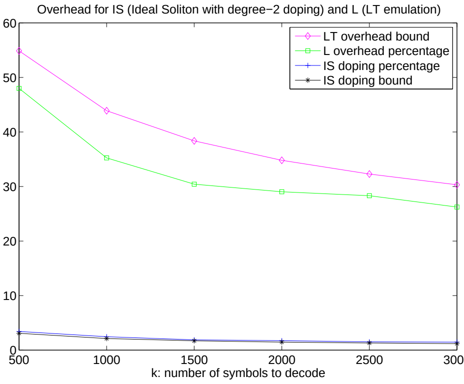

Fig. 3. Overhead (doping) percentage: we define s_t as the number of symbols collected in both collection phases, and the collection overhead ratio as $(s_t - k)/k$, which allows us to compare the overhead for the LT decoding of k original symbols and IS, the degree-two doped belief-propagation decoding of k coded symbols with Ideal Soliton degree distribution. Finally, we show the analytical bound obtained for IS doping percentage, based on (16)

$$p_t = [0\ 0\ 1\ 0 \cdots 0]\, \mathbf{P}^t\, [1\ 0\ 0 \cdots 0]^T . \tag{12}$$

After a series of matrix transformations, we obtain the following recursive expression for the probability of stopping at time $t + 1$

$$\Pr\{t+1\} = \eta(0) \left(\wp^{(t)}(t-1) - \sum_{i=1}^{t-1} \Pr\{t-i\}\, \wp^{(i)}(1+i) \right), \tag{13}$$

where $\wp^{(s)}(d)$ denotes Poisson distribution of intensity s, evaluated at d. We calculate the expected value of the interdoping yield Y as

$$E[Y] \approx \sum t \Pr\{t\}. \tag{14}$$

where the approximation is accurate to the extent that the model in (5) is accurate and that the stopping time probability computation in (13) is independent of the change in the decoding iteration count ℓ.

Recall that, due to doping, the decoding process is a renewal process, and thus, the total duration D_h of the decoding process up to hth doping is the partial sum of h epochs, modeled by interdoping yields (IID random variables $Y_i = T_i - T_{i-1}$). Hence, $D_h = \sum_{i=1}^{h} Y_i$. The expected number of dopings sufficient for complete decoding is equivalent to the stopping time of the random walk D_h, where the stopping threshold is k. The total number od dopings is the stopping time random variable D defined as

$$D = \min \{ h : D_h \geq k \} . \tag{15}$$

Hence, since the interdoping yield Y is a non-zero-mean random variable, the Wald Equality [6] implies that the mean stopping time is

$$E[D] = k / E[Y] . \tag{16}$$

B. Cost Comparison

Our analysis is based on the Ideal Soliton for tractability reasons. However, the simulation results show that this dis-

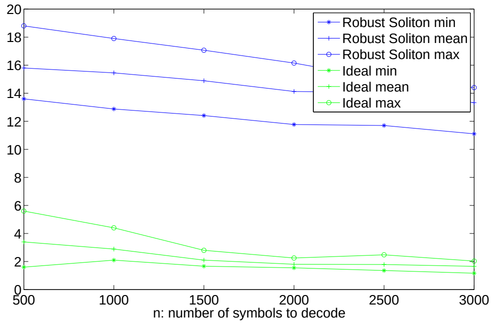

Fig. 4. Doping percentage with Ideal Soliton degree distribution vs Robust Soliton.

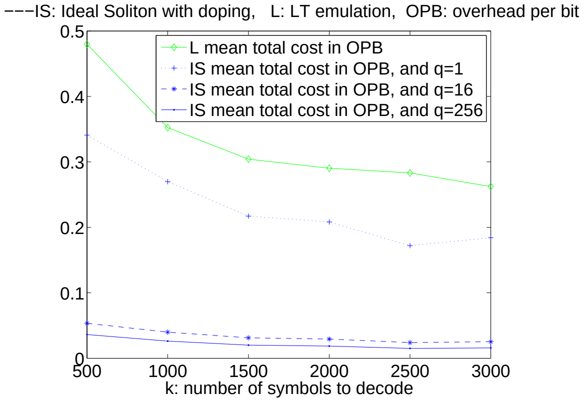

Fig. 5. Cost Comparison (in overhead per bit) of Ideal Soliton based Fountain decoding with ARQ and doping versus the emulated cost of LT codes, for $q \in 1, 16, 256$ bits and $\delta = 0.01$. The overhead plotted is the actual cost s_t decreased by qk, which is the minimum cost needed for decoding, then divided by qk.

tribution is a better choice for an ARQ-enhanced decoder than the Robust Soliton, and that it indeed provides the expected cost savings in combination with ARQ. Figure 3 illustrates the dramatic cost overhead $(s_t - k)/k$ reduction when employing doping with an Ideal Soliton distribution relative to the overhead of Robust Soliton encoding without doping. Note that here $s = k$. Figure 4 demonstrates that Robust Soliton with doping performs markedly worse than Ideal Soliton encoding. And, in particular, it illustrates that Ideal Soliton with doping demonstrates a very low variance which is surprisingly different from the results without doping.

For the range of k values presented in Figure 3, and Ideal Soliton based codes, the results obtained analytically through (13), (14) and (16) with the expectation sum in (14) computed up to $n_2 = k/2$ (hence neglecting the decoding/coping count, i.e., assuming $k \gg \ell$) are close to the ones obtained using simulation results obtained without any approximations.

The total cost of decoding (in bits) for $s = k$ is

$$s_t = qk + \frac{k(q + \log k)}{E\,[Y]}. \tag{17}$$

In Figure 5, we present an alternative measure of cost, referred to as the overhead per bit (OBP) o_t, where $o_t = (s_t - qk)/qk$. This measure is more convenient to express throughput benefits of doped decoding, as $\chi = 1/(1 + o_t)$.

IV. Conclusion

In this work, we present a BP decoding scheme for Fountain codes that leverages some nice properties of Ideal Soliton distribution by employing an ARQ mechanism to dope the decoding process whenever it gets stalled. We introduce an approximate random walk model of the BP decoding ripple evolution process, and study the throughput improvements over the classical BP decoder. The doping bound based on the random walk stopping time analysis gives a close prediction of a set of simulation results for a fountain code based on an Ideal Soliton distribution. The elegant form of our random walk stopping time result that avoids matrix multiplication, is a contribution by itself, as it streamlines numerical calculations which is particularly important for large number of input symbols k. Ideal Soliton distribution significantly outperforms the Robust Soliton employed with LT codes when employed with the proposed random two-degree doping ARQ mechanism.

There may be other benefits of the proposed decoding scheme. An application of doped belief-propagation decoding with ARQ to network-coded Fountain codes is described in [4]. This application aims to achieve minimum-delay data collection from a specific wireless sensor network, dubbed *circular squad network*, and offers an analysis of the impact the coded symbol degree distribution and the doping mechanism have on the collection delay.

Our future work will focus on extending the model by dropping the assumption on the uniqueness of ripple symbols and including the effects of the decoding/doping iteration count. Apart from that, we plan to replace the assumption that the number of initially transmitted (before doping) coded symbols is $s = k$, with a more general assumption that $s = k\,(1 + \delta)\,$, where $0 < \delta << 1$.

References

[1] G. Caire, S. Shamai, and S. Verdu. Feedback and belief propagation. In *4th Int. Symp. on Turbo Codes and Related Topics*, 2006.

[2] E.Maneva and A.Shokrollahi. New model for rigorous analysis of lt-codes. In *ISIT 2006*, 2006.

[3] R. Karp, M. Luby, and A.Shokrollahi. Finite length analysis of lt codes. In *ISIT 2004*, 2004.

[4] S. Kokalj-Filipovic, P. Spasojevic, Roy Yates, and Emina Soljanin. Decentralized fountain codes for minimum-delay data collection. In *CISS 2008*. IEEE, March 2008.

[5] M. Luby. L t codes. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[6] R.Gallager. *Discrete Stochastic Processes*. Kluwer Academic Publishers, 1995.

[7] S. Sanghavi. Intermediate performance of rateless codes. In *Information Theory Workshop, 2007. ITW '07*, 2007.

[8] A. Shokrollahi. Raptor codes. *IEEE Trans. Inform. Theory*, 52, Jun 2006.

[9] E. Soljanin, N. Varnica, and P. Whiting. Incremental redundancy hybrid arq with ldpc and raptor codes. *IEEE Trans. Inform. Theory, submitted*, Sep 2005.

[10] S. ten Brink. Code doping for triggering iterative decoding convergence. In *International Symposium on Information Theory*, 2001.