The one-out-of-k retrieval problem and Linear Network Coding

Giuseppe Bianchi, Lorenzo Bracciale, Keren Censor-Hillel,
Andrea Lincoln, Muriel Médard

Abstract In this paper we show how linear network coding can reduce the number of queries needed to retrieve *one specific message* among k distinct ones replicated across a large number of randomly accessed nodes storing one message each. Without network coding, this would require k queries on average. After proving that no scheme can perform better than a straightforward lower bound of $0.5k$ average queries, we propose and asymptotically evaluate, using mean field arguments, a few example practical schemes, the best of which attains $0.82k$ queries on average. The paper opens two complementary challenges: a systematic analysis of practical schemes so as to identify the best performing ones and design guideline strategies, as well as the need to identify tighter, nontrivial, lower bounds.

Key words: Delay Tolerant Network, Linear Network Coding, Fluid Approximations

1 Introduction

This paper introduces a new problem, which we call *one-out-of-k retrieval*. Supose there are k distinct messages $X = \{x_1, ..., x_k\}$, where $x_i \in 0, 1^m \; \forall i \in [1, k]$. A receiver wishes to learn all m bits of one *specific* target message, $x_r \in X$. We can produce some new set of messages $Y = \{y_1, y_2, ...\}$ of arbitrary size and contents. Each round, the receiver can request a message selected over a pre-determined probability distribtution from Y. We wish to come up with a set of linearly coded messages for Y and a probability distribution over these such that the average number of rounds in which a message must be requested by the receiver from Y to learn all m bits of x_r is minimized.

This scenario is practically encountered in Delay Tolerant Networks (DTN). In such networks, data replication across the moving terminals is at the core of most proposed data access or data delivery solutions, as the likelihood that a user interested in a specific data item "physically" meets only the single data producer becomes rapidly negligible as the network size scales.

Contribution

Most network coding research has focused on retrieving and decoding *all* the messages instead of a specific subset Even for the case of $k = 2$, schemes exist which take an average of ≈ 1.828 coded messages from Y, outperforming the naive average of 2. This raises some questions: how much reduction in average numbers of messages from Y can we gain? And with which practical constructions?

In the paper, we present a lower bound of $0.5k$ for the average number of rounds the receiver must request messages. Then, we propose some initial example schemes where the selection of the probability distribution over Y results in a lower average number of requests than the naive average of k messages needed from the set Y.

Moreover, we provide a general methodology to analyze such schemes. We specifically show how to apply mean field arguments to derive the asymptotic performance of the proposed approaches. We concretely apply our methodology to two example schemes, the best of which attains an average of $0.82k$ rounds of communication.

Previous Work

Previous work on network coding in DTNs has not considered the problem of solving for *one* out of k messages. In our model, the protocol does not allow for the receiver to request the specific information it wants and nor do we treat it as wanting all information. For instance, LT codes [6] are designed with the different goal of optimizing the decoding procedures. Many papers [8], [7], [10], [3] investigate routing protocols in DTNs. These papers attempt to decode all messages , as opposed to just *one* of k. Yoon and Hass consider application of linear network coding to DTNs but, unlike this work, investigate the case of sparse networks [9].

2 Network Model and Problem Statement

In our model there are k messages $X = \{x_1, ..., x_k\}$, each of which is a can be represented by a binary vector of length m bits. There is a receiver node, r, which wants to know the contents of the one message, we will call this message x_r. The receiver, r, travels throughout the network and will receive messages from the nodes it contacts in close proximity. We model this as r contacting a random node, which transmits its output. These contacts cannot be commanded so messages may be repeated and r can not query for a particular message. In each round, the receiver node r receives exactly one coded message, y, from one of the transmitting nodes. Each round has a constant duration. The nodes in this network can store linear combinations of messages over some field F_f.

Definition 1. The type (or degree) of a coded message is the number of message linearly combined in that data message.

These linear combinations are stored with header data that specifies which messages were summed with what multiplicative constants.

Definition 2. Solving for message x_j means determining all m bits in the message x_j.

Definition 3. The *one-out-of-k* retrieval problem is determining what coding scheme produces the lowest expected time for r to solve for x_r where a coding scheme is the proportion $p_1, p_2...p_k$ of the codeword degrees distributed in the networks.

In other words, we want to find $p_1 \cdots p_k$ that minimize the time for retrieving only one message, given that the receiver collects at each round an uncoded message with probability p_1, a "pair" (codeword with degree 2) with probability p_2, a "triplet" with probability p_3 etc.

Thus, Y is the set of all linear combinations of the k messages in X. Each coded message $,y \in Y$, is a linear combination of n messages and has a probability $\frac{p_n}{\binom{k}{n}}$ of being sent to the receiver.

2.1 A trivial example: $k = 2$

Consider the simple case where we have only two kind of different message that we call A and B. If we do not use coding ($p_1 = 1, p_2 = 0$) it is trivial to show that the average time spent from the receiver for collecting A (or equivalently B) is 2, i.e. k. Similarly if all nodes carry a random linear combination of both A and B ($p_1 = 0, p_2 = 1$) the expected retrieval time is *exactly* 2 encounters, so once again the average is 2. Now let AB be the linear combination of A and B so that at each encounter the receiver can collect A with probability $p/2$, B with probability $p/2$, and AB with probability $1 - p$. The average delay to retrieve item A is:

$$\text{Delay} = 1 - p + \frac{1}{1 - p/2}$$

Then it is trivial to show that when $p = 2 - \sqrt{2}$ the expected time to retrieve A is minimized and equals to $2\sqrt{2} - 1 \approx 1.828$, i.e. about 9% lower than both previous cases. Hence this problem is solved adopting the coding scheme $p_1 = 2 - \sqrt{2}, p_2 = \sqrt{2} - 1$. The delay versus p is shown in figure 1.
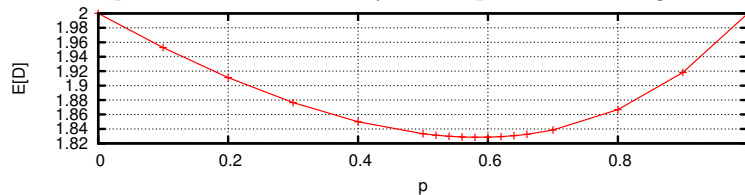

Fig. 1 Average retrieval delay for the case of $k = 2$

2.2 Lower bound

For the problem of determining the contents of one message out of k we prove that $0.5k$ messages is the lowest achievable average cost.

Intuitively level to solve for c messages we must receive at least c coded messages.

Lemma 1. *On average there are greater than $\frac{1}{2}k$ messages solved for before or in the same round as x_r.*

Proof. First let us define m_j as the number of messages solved before or at the same time that x_j is solved. If a message x_j never has its contents solved then define $m_j = k$. For convenience, let m_r be the number of messages solved before or at the same time as the message of interest x_r. x_r is randomly selected from $\{x_j | j \in [1, k]\}$. Thus, the average number of message solved before or at the same round as x_r is the average vale of m_j i.e. $\frac{\sum_{i=1}^{k} m_j}{k}$.

If all the values of m_j are distinct then the minimum value they can have is the integers from 1 to k thus the average value of m_j is:

$$\frac{(k+1)k}{2k} = \frac{k+1}{2}.$$

If some messages are solved at the same time (in the same round) then this sum is strictly greater because having multiple messages solved at the same time causes double counting.

Thus, on average there are greater than $\frac{1}{2}k$ messages solved before or in the same time step as x_r. ∎

Next we use this lemma to prove a lower bound on the average number of rounds needed.

Theorem 1. *There exists no scheme in our model such that the contents of a message x_r selected at random can be solved with fewer than $\frac{1}{2}k$ coded messages on average.*

Proof. Given lemma 1 when the receiver, r, has solved for x_r, having received t_r coded messages, r has also solved for more than $\frac{1}{2}k$ messages.

To solve for m_j messages the reciever must receive at least m_j coded messages. Thus the average number of coded messages needed to solve for x_r must be greater than or equal to the average value of m_j. Thus a lower bound for the average number of coded messages needed is $k/2$. ∎

3 Methodology

Determining whether the set of received messages fully specifies the target *one-out-of-k* message, is the major difficulty. Since messages are retrieved at random, differently coded messages are collected (e.g. uncoded messages, linear combination of two messages, linear combination of all k messages,

and so on depending on the construction). The set of collected messages also depends on time, requireing a *transient* stochastic process to model a chosen strategy, which usually exhibits a non-trivial space state.

To avoid such stochastic modeling complexity, the methodology employed hereafter consists of three steps: i) model a proposed coding strategy via a discrete time (vector) stochastic process; this is arguably the most complex step, as discussed later on; ii) approximate the proposed coding strategy's transient solution with the deterministic mean trajectory specified by the drift (vector) differential equation of a conveniently rescaled stochastic process, and iii) derive the average number of queries needed to retrieve the target message from a relevant probability distribution, which is derived from the knowledge of the drift equation solutions.

The approximation in step (ii) above is motivated by by the fact that practical values of k are relatively large. It consists of using mean field techniques widely established in the literature since [5], which have been successfully applied to a variety of problems [2, 4], and which guarantee asymptotic convergence to *exact* results for finite state space systems under mild assumptions (see e.g., [4]). Our own results show a very accurate matching with simulation even for relatively small values of k.

Details and a simple example of the proposed methodology are presented in appendix 1 of the technical annex [1].

4 Practical Example Cases

In order to understand the *asymptotic* nature of the gain, and show how the proposed methodology can be concretely applied we show two example constructions. In both cases, we compare analytical results with simulation.

All-or-nothing scheme

This scheme is extremely simple in terms of states, permits a simple analysis, and can be used as a reference to gauge the improvements brought about by more complex schemes. The *all-or-nothing* scheme comprises only two possible types of messages, defined below.

Definition 4. A *singleton* is a message x_i for $i \in [1, k]$ sent in plain text.

Definition 5. A *fully coded message* is a random linear combination $\sum_{i=1}^{k} \alpha_i x_i$ of all k messages over a large field size $\mathbb{F}$, with $\alpha_i \in \mathbb{F}$.

We assume that all messages x_i, with $i \in [1, k]$, are equiprobable. Under this assumption, the *all-or-nothing* scheme is characterized by a single parameter p, where p is the singleton reception probability and $1 - p$ is the complementary fully coded message reception probability. The state space thus comprises two state variables: i) the number of singletons received at a given time, and ii) the number of fully coded messages received at the same time.

Theorem 2. *The* all-or-nothing *scheme achieves a best possible performance of* $0.86k$; *which corresponds to the value* $p \approx 0.6264$.

Proof. Using the methodology presented above, let we define the following two density processes:

- $s(t) \in (0, 1)$ is the fraction of singletons accumulated until time t;
- $d(t) \in (0, 1)$ is the fraction of fully coded messages accumulated until time t.

In this case, the drift differential equation reduces to two independent ordinary differential equations. For the case of singletons, operating in a similar way to the example in Appendix 1 of [1], we have:

$$s'(t) = 1 - ps(t), \tag{1}$$

which, when solved with initial conditions $s(0) = 0$, yields

$$s(t) = 1 - e^{-pt}. \tag{2}$$

For the case of fully coded messages, we have:

$$d'(t) = (1 - p)d(t) \quad \text{with } d(0) = 0. \tag{3}$$

Therefore
$$d(t) = (1 - p)t. \tag{4}$$

We now note that a target message is decoded when either the corresponding singleton is received, or when the number of received singletons plus the number of fully coded messages is equal to the total number k of distinct messages. In terms of density processes, this latter condition is expressed by the equation
$$s(t) + d(t) = 1 \qquad \rightarrow \qquad e^{-pt} + t = 1. \tag{5}$$
Let us call t^* the solution of this transcendental equation. By introducing the Lambert W function, we can express t^* in closed form as $t^* = \frac{W(\frac{p}{1-p})}{p}$.

Finally, the average number of messages $E[X]$ needed to decode the target message can be computed:

$$E[X] = \int_0^{t^*} s(\tau)d\tau = \frac{1 - e^{-W\left(\frac{p}{1-p}\right)}}{p} \tag{6}$$

This expression is minimized when $p = 0.626412$, and yields a minimum (normalized) number of retrieved messages $E[X] = 0.859884$. ∎

In order to verify the correctness of the analysis, Figure 2-a shows that simulations vary the number of messages from k=2 to k=70. Note that the theoretical results have an asymptotic nature, hence our choice of running

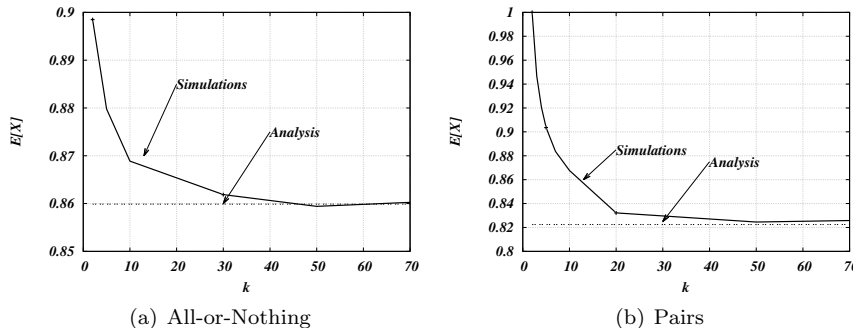**Fig. 2** Average retrieval delay varying the number of messages: mean field approximation vs simulation.

simulations with small values of k. Every point in the figure is the delay to retrieve a data message averaged on 50000 samples. Even though the proposed methodology obtains an exact solution only for large values of k, already after k=20 the error is below 1%.

Pairs-only scheme
This scheme shows how the state space can become extremely complex (actually an infinite set of state variables) even when considering an apparently very simple approach. Moreover, it can be solved using an alternative methodology, because its emerging decoding structure can be cast as an Erdös-Rényi random graph; thus it permits us to verify that our methodology, despite being extended to the case of infinite state variables (hence violating the assumptions in [4]), nevertheless yields the same results derived in the relevant random graph literature.

As the name suggests, the *pairs-only* scheme includes only one type of coded message, namely the random linear combination of two randomly chosen messages. This type of message is called *pair* and is formally defined as follows.

Definition 6. A *pair* is a random linear combination of two randomly chosen messages over a large field size in the form $\{(\alpha x_i + \beta x_j)|i \neq j \text{ and } i,j \in [1,k]\}$ where $\alpha, \beta \in \mathbb{F}$ and $\mathbb{F}$ is a large field.

In analyzing this scheme, the difficulty lies in defining an appropriate state space. Once this is done, the remaining analysis reduces to the conceptually straightforward application of our methodology. The state space definition and justification is presented in Appendix 2 [1], along with the proof of the following theorem:

Theorem 3. *The* pairs-only *scheme achieves a performance of* $\frac{\pi^2}{12}k \approx 0.8224k$.

Our results confirm those found in random graphs literature. However, our approach can be extended to coding schemes which cannot be directly cast as

a random graph problem, such as, the combination of singletons and pairs, which yields a performance slightly below $0.8k$ (we postpone analysiss to a later extended version of this work). Comparison with simulation results averaged over 50.000 realizations is reported in Figure 2-b. Again, results show that convergence to the asymptotic result is very fast, with an error lower than 1% for $k > 20$.

5 Conclusion

In this work we explore efficient solutions to *one-out-of-k retrieval*.We prove a lower bound of $0.5k$ and upper bound of $0.8224k$ on the number of coded messages needed on average to solve for the message of interest. Current simulation results suggest that the true minimum value for *one-out-of-k retrieval* should be higher than $0.5k$. The machinery given in Section 3 can be used to analyze various proposed schemes to produce upper bounds. Generalizing *one-out-of-k retrieval* to *m-out-of-k retrieval* is another interesting extension.

References

1. Bianchi, G., Bracciale, L., Censor-Hillel, K., Lincoln, A., Médard, M.: Technical annex. URL: `http://netgroup.uniroma2.it/docs/tech_annex.pdf`
2. Buckley, F.M., Pollett, P.K., et al.: Limit theorems for discrete-time metapopulation models. Probability Surveys **7**, 53–83 (2010)
3. Chen, L., Ho, T., Low, S., Chiang, M., Doyle, J.: Optimization based rate control for multi-cast with network coding. In: Proc. IEEE Infocom (2007)
4. Darlings, R.W.R., Norris, J.R.: Differential equation approximations for markov chains. Probability Surveys **5**, 37–79 (2008)
5. Kurtz, T.G.: Solutions of ordinary differential equations as limits of pure jump markov processes. Journal of Applied Probability **7**(1), 49–58 (1970)
6. Luby, M.: Lt codes. In: Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on, pp. 271–280 (2002). `doi:10.1109/SFCS.2002.1181950`
7. Sassatelli, L., Medard, M.: Inter-session network coding in delay-tolerant networks under spray-and-wait routing. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks **10**, 103 – 110 (2012)
8. Widmer, J., Le Boudec, J.: Network coding for efficient communication in extreme networks. ACM SIGCOMM workshop on Delay-tolerant networking pp. 284–291 (2005)
9. Yoon, S., Haas, Z.: Application of linear network coding in delay tolerant networks. IEEE Ubiquitous and Future Networks (ICUFN) pp. 338–343 (2010)
10. Zhang, X., Neglia, G., Kurose, J., Towsley, D.: On the benefits of random linear coding for unicast applications in disruption tolerant networks. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks **4**, 1–7 (2006)