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, non trivial, lower bounds.

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

1 Introduction

This paper introduces a new problem, that we call *one-out-of-k retrieval.* This problem, in its abstract form, can be stated as follows. Assume k distinct messages are replicated across a large (infinite) number of nodes storing one message each; a client, wishing to retrieve one *specific* target message among such k ones, has no means to a priori know on which nodes the target message is stored, nor can control or enforce the order in which nodes will be queried. How many queries, on average, are needed to retrieve the target message?

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 an user interested in a specific data item ”physically” meets only the single data producer becomes rapidly negligible as the network size scales.

Moreover, a distinguishing aspect of DTNs is that data exchange among two moving devices can occur only when the devices get in proximity each other, so that a short-range wireless connection (e.g. Bluetooth or WLAN) can be established. In such scenario, the time needed to download a data content from a neighbour device can be order of seconds, and, assuming a given average inter-contact time, the retrieval time is dominated by (and can be measured in terms of) the number of contacts elapsed before a device storing a looked for data item is encountered.

Contribution

Perhaps surprising even when $k = 2$ naive schemes take an average of 2 encounters but can be improved to ≈ 1.828. This observation has been apparently neglected by network coding research, which has mainly focused on retrieving and decoding *all* the data blocks/items instead of a specific one, and has used uncoded data messages or small combinations of data items (e.g. Luby codes) for improving the decoding complexity, but still in the context of decoding all the data rather than one specifically targeted item. It opens the room for a plethora of novel questions: is this apparent average retrieval gain vanishing away with a larger number of items or has an asymptotic nature? How much can we gain? And with which practical constructions?

Goal of this work is to raise the attention of the network coding community on such problem, as well as provide some preliminary steps towards its understanding. This work is unique in that we bound the minimum time to receive one out of k messages, as opposed to bounding the time to receive all k messages. More specifically, in the paper, after proving a straightforward lower bound of $0.5k$ (annex Appendix 4 [3]), we propose some initial example schemes, which involve mixing coded messages that are formed from linear combinations of different numbers of messages. From the perspective of engineering coding schemes, mixing message coding types is a contribution of this work.

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 $0.82k$. Definitely, there is still room for gain as our research so far has not yet targeted exhaustiveness; however, the fact that all the schemes proposed are largely above the straightforward $0.5k$ lower bound suggests that much tighter lower bounds may be found.

Previous Work

Previous work 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 [8] are designed with the different goal of optimizing the decoding procedures. Many papers [4], [6], [10], [7] 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 paper, investigate the case of sparse networks [9].

2 Network Model and Problem Statement

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 bumping into contacting a random node, which transmits its output. These contacts can't be ordered 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. The nodes in the DTN can store linear combinations of messages over some field F_f. These linear combinations are stored with header data that specifies which messages were summed with what multiplicative constants.

Definition 1. Solving for message x_j means determining all the m bits in the message x_j.

The problem is determining what coding scheme produces the lowest expected time for r to solve for x_r where a coding scheme is a set of linear combinations of messages $Y = y_1, y_2, ...$ and the associated proportion of transmitting nodes on which these linear combinations are stored $p_1, p_2....$ For an example of this analysis performed for $k = 2$ look at annex Appendix 1 [3].

3 Methodology

The major hurdle emerging in the evaluation of practical algorithms consists in the need to determine whether the so far received set of messages is sufficient to decode the target *one-out-of-k* message. 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), and the set of collected messages depend on time, thius requires us to model a chosen strategy as a *transient* stochastic process, which usually exhibits a non trivial space state.

To at least in part circumvent such stochastic modeling complexity, the methodology employed hereafter consists in three steps: i) model a proposed coding strategy via a a discrete time (vector) stochastic process; this is arguably the most complex step, as discussed later on; ii) approximate its 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, in turns derived from the knowledge of the drift equation solutions. Approximation (ii) above is motivated by by the fact that practical values of k are relatively large. It consists in casting to our needs mean field techniques widely established in the literature since [5], which which have been successfully applied to a wide set of problems [1, 2], and which guarantee asymptotic convergence to *exact* results for finite state space systems under mild assumptions, see e.g., [2]. Our own results will indeed show a very accurate matching with simulation even for relatively small values of k, as low as some tens.

More formally, let's assume a discrete time scale, clocked by message arrivals, i.e., time $n \in \{1, 2, \cdots\}$ is defined as the time of arrival of the n-th element. Let us now identify a model for the *receiver state*. This is a critical step (as will appear in the construction examples discussed appendix 5 [3]), as the relation between receiver state and the different "types" of messages collected (and how many) is in general not trivial and specific for every scheme considered; For instance, the reception of two different "types" of message, say a linear combination of messages A and B (later on called "pair"), and a message A (later on called "singleton") yields the decoding of message B, and suggests to use as state variables the number of message "types" resulting *after* decoding, in this case the two singletons A and B, rather than the actually received message types (a pair an a singleton).

In most generality, the status of the receiver at an arbitrary discrete time n is summarized by means of a state vector:

$$\bar{\psi}(n) = \{\psi_1(n), \psi_2(n), \cdots\} \tag{1}$$

where $\psi_i(n)$ is defined as the number of messages of "type" i stored by the receiver at time n.

Under the assumption of independent random messages being retrieved at each time step, and appropriate choice of the space state, $\bar{\psi}(n)$ introduced in (1) is a discrete-time Markov chain. Let us now now write the relevant time-dependent state transition probabilities as functions of the vector state components normalized with respect to k, i.e.:

$$P\left\{\bar{\psi}(n+1)|\bar{\psi}(n)\right\} = f_{\bar{\psi}(n+1)}\left(\frac{\bar{\psi}(n)}{k}\right) \tag{2}$$

The conditional expectation, namely the *drift* of the considered Markov chain, is readily given by the vector

$$E\left[\bar{\psi}(n+1) - \bar{\psi}(n)|\bar{\psi}(n)\right] = \sum_{\bar{v} \in \text{all states}} \left(\bar{v} - \bar{\psi}(n)\right) f_{\bar{v}}\left(\frac{\bar{\psi}(n)}{k}\right) = \bar{d}\left(\frac{\bar{\psi}(n)}{k}\right), \tag{3}$$

where we conveniently express the state vector components as normalized with respect to k. We now introduce a new stochastic process which is a *doubly-rescaled* version of (1) in terms of both state (normalized with respect to k, i.e., a *density process* [1]) as well as time (also normalized with respect to k, i.e. $t = n/k$):

$$\bar{\sigma}(t) = \frac{\bar{\psi}\,(t \cdot k)}{k}$$

The conditional expectation (3) is readily rewritten for the rescaled process as:

$$E\left[k \cdot \bar{\sigma}(t+1/k) - k \cdot \bar{\sigma}(t) | \bar{\sigma}(t)\right] = \frac{E\left[\bar{\sigma}(t+1/k) - \bar{\sigma}(t) | \bar{\sigma}(t)\right]}{1/k} = \bar{d}\,(\bar{\sigma}(t)) \quad (4)$$

For large k, and under quite general assumptions (it suffices the drift $\bar{d}(.)$ to be a Lipschitz vector field [2]), the density process $\bar{\sigma}(t)$ converges in probability to a deterministic trajectory, computed by solving the system of differential equations obtained by replacing the left side of equation (4) with the derivative $\bar{\sigma'}(t)$:

$$\bar{\sigma'}(t) = \bar{d}\,(\bar{\sigma}(t)) \quad (5)$$

at last, from the knowledge of $\bar{\sigma}(t)$, the average number of messages needed to decode the target message is readily computed.

In order to better clarify, we present a trivial example in appendix 5 [3].

4 Practical Example Cases

In order to understand the *asymptotic* nature of the gain, and show how the proposed methodology can be concretely applied, in this section, with no pretense of systematic exploration, we show two example constructions. In both cases, we compare analytical results with simulation.

All-or-nothing scheme

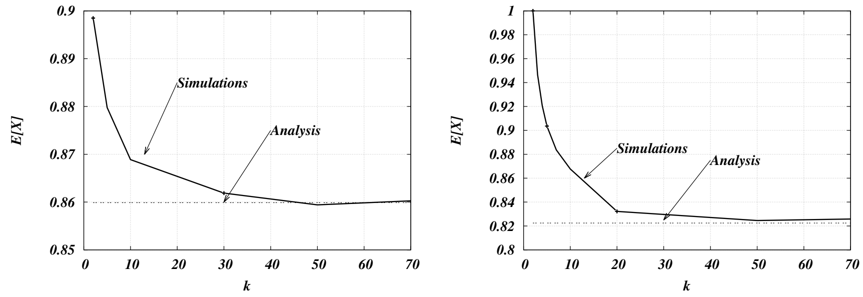
This scheme is selected as it is extremely simple in terms of states, permits a simple analysys, and can be used as a reference to gauge the improvements brought about by more complex schemes. The *all-or-nothing* scheme comprises only two possible types of messages, defined below.

Definition 2. A *singleton* is a message x_i for $i \in [1, k]$ sent in plain text.

Definition 3. 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, the singleton reception probability, whereas $1 - p$ is the complementary probability that a node receives a fully coded message. 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 1. *The* all-or-nothing *scheme achieves a best possible performance of* $0.86k$*; this corresponds to the value* $p \approx 0.6264$.



(a) All-or-Nothing (b) Pairs

Fig. 1 Average retrieval delay varying the number of messages: analysis vs simulation.

The proof is in annex Appendix 2 [3]. In order to verify the correctness of the analysis, Figure 1-a shows simulations varying the number of messages from k=2 to k=70. Note that the theoretical results have an asyptotic nature, hence our choice of running simulations with small small values of k. Every point in the figure is the delay to retrieve a data message averaged on 50000 samples. Even if the proposed methodology attains an exact solution for large values of k, already after k=20 the error is below 1%.

Pairs-only scheme

This scheme is selected as it shows how the space state can become extremely complex (actually an infinite set of state variables) even when considering an apparently very simple approach. Moreover, it is chosen because it could be solved using an alternative methodology. Indeed its emerging decoding structure can be cast as an Erdos-Renyi random graph; thus it permits us to verify that our methodology, despite being extended to the case of inifinite state variables (hence violating the assumptions in [2]), 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 4. 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 real hurdle is to define an appropriate state space; once this is done, the remaining analysis reduces to the conceptually straightforward application of our methodology (although some non trivial calculus will be actually needed to solve the drift differential system, as detailed in the annex Appendix 3 [3]). State space definition and justification is presented in annex Appendix 3 [3], along with the proof of the following theorem:

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

Our results, obtained with a different approach, indeed 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, for instance, the combination of singletons and pairs (which yields a performance slightly below $0.8k$, but which we postpone, for space reasons, to an extended version of this work). Comparison with simulation results averaged over 50.000 realizations is reported in Figure 1-b. Again, results show that convergence to the asymptotic result is very fast, with an error lower than 1% already at $k > 20$.

5 Conclusion

In this paper we explore solutions efficient solutions to *one-out-of-k retrieval.* This paper proves 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. It appears from current simulation results that the true minimum value for *one-out-of-k retrieval* will be higher than $0.5k$. The machinery given in Section 3 can be used to analyze various proposed schemes to produce upper bounds. A proposed extension of this paper an upper bound of lower than 0.8 would be presented. In general tightening the lower and upper bounds is an open problem. Generalizing *one-out-of-k retrieval* to *m-out-of-k retrieval* is another interesting extension.

Acknowledgements This reseach was funed in part by the Bounded-Contention Coding grant.

References

1. Buckley, F.M., Pollett, P.K.: Limit theorems for discrete-time metapopulation models. Probability Surveys **7**, 53–83 (2010)
2. Darlings, R.W.R., Norris, J.R.: Differential equation approximations for markov chains. Probability Surveys **5**, 37–79 (2008)
3. Giuseppe Bianchi, Lorenzo Bracciale, K.C.H.A.L.M.M.: Technical annex. URL: `http://netgroup.uniroma2.it/docs/tech_annex.pdf`
4. J Widmer, J.L.B.: Network coding for efficient communication in extreme networks. ACM SIGCOMM workshop on Delay-tolerant networking pp. 284–291 (2005)
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. L Sassatelli, M.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)
7. Lijun Chen, Tracey Ho, S.L.M.C.J.D.: Optimization based rate control for multi-cast with network coding. In: Proc. IEEE Infocom (2007)
8. 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`
9. SK Yoon, Z.H.: Application of linear network coding in delay tolerant networks. IEEE Ubiquitous and Future Networks (ICUFN) pp. 338–343 (2010)
10. Xiaolan Zhang, G. Neglia, J.K.D.T.: 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)