Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering

Supratim Deb, Muriel Médard, and Clifford Choute
Laboratory for Information & Decison Systems,
Massachusetts Institute of Technology
{supratim,medard,choute}@mit.edu

Abstract

We study the problem of simultaneously disseminating multiple messages in a large network in a decentralized and distributed manner where nodes only have knowledge about their own contents. We consider a network with n nodes and k messages spread throughout the network to start with, where not all nodes have all the messages. Our communication model is such that the nodes communicate in discrete-time steps, and in every time-step, each node communicates with a *random* communication partner chosen uniformly from all the nodes (known as the *random phone call* model). In each time-step, only one message can be transmitted. The goal is to rapidly disseminate all the messages among all the nodes. We study the time required for this dissemination to occur *with high probability*, and also in expectation.

We show that a *random linear coding* (RLC) based protocol for message dissemination disseminates all the messages among all the nodes in time $ck + O(\sqrt{k}\ln(k)\ln(n))$ for a suitable constant $c > 0$. Analytical results show that, $c < 3.46$ using *pull* based dissemination and $c < 5.96$ using *push* based dissemination, but reported simulations suggest that $c < 2$ might be a tighter bound. Thus, if $k \gg (\ln(n))^3$, the time required to simultaneously disseminate the messages using RLC is asymptotically at most ck which is a substantial improvement over disseminating the messages sequentially (i.e., one after the other) that takes a time of at least $k\log_2(n)$. Furthermore, in the regime $k \gg (\ln(n))^3$, the dissemination time is clearly order optimal, since disseminating k message takes at least k rounds. In the regime $k \ll (\ln(n))^2$, the dissemination time with RLC goes down by a factor of $\Omega(\sqrt{k}/\ln k)$ (as opposed to $\Theta(\ln(n))$ in the regime $k \gg (\ln(n))^3$) as compared to sequential dissemination. The cost of the RLC protocol is an overhead associated with every transmission, but the overhead is negligible for messages of reasonable size. We also consider a *store and forward* mechanism without coding, which is a natural extension of *gossip-based* dissemination with one message in the network. We show such an

approach performs badly for large values of k, i.e., when $k = \alpha n$ for some $\alpha \leq 1$, and does no better than a sequential approach (instead of doing it simultaneously) of disseminating the messages which takes $\Theta(k \ln(n))$ time.

While the asymptotic results might sound believeable based on the dissemination protocol using RLC, owing to the distributed nature of the system, the proof requires careful modeling and analysis of an appropriate time-varying Bernoulli process.

1 Introduction

Of late, design of protocols to rapidly disseminate messages in large networks have gained a lot of attention. Much of the research on information dissemination started with updating messages in large distributed computer systems where computers can communicate with each other. More recently, the emergence of sensor networks has added a new paradigm to the problems of distributed message dissemination.

The use of *gossip-based* protocols to disseminate message was first proposed in [4]. In *gossip* based protocols, nodes communicate with each other in communication steps called *rounds*, and the amount of information exchanged in each round between two communicating nodes is limited. Further, there is no centralized controller and every node in the network acts simply based on *state or information* of the node, and not that of the over all network. Thus, *gossip* based protocols are inherently distributed and easily implementable, and provides powerful alternatives to *flooding* and *broadcast* based protocols for dissemination of messages. There is a wide literature on the practical aspects of gossip-based message dissemination [16]. A detailed analysis of a few gossip-based dissemination schemes were provided in [11]. A common performance measure in all the previous work is the time required to disseminate a single message to all the nodes in the network. In recent work, [13, 12] considered the scenario where there are multiple messages, each with a unique destination. The authors considered what they call *spatial gossip* where the nodes are embedded in a metric-space with some distance metric between the nodes. More recently, in [3], the authors have studied gossip-like distributed algorithms for computing averages at the nodes. In the framework they consider, each node starts with possibly distinct value of a certain parameter, and the goal is to find the time required for every node to compute the average of the parameter values at the nodes, using gossip-like algorithms. In this paper, we envisage a different problem, in which the network seeks to simultaneously disseminate multiple messages to all the nodes in a distributed and decentralized manner. Thus, each node wants to compute the exact messages at every other node.

As pointed out in [13], we note that, any *gossip* protocol has two layers of design aspects. One is the design of *gossip algorithm* by which, in every round, every node decides its communication partner, either

in a deterministic, or in a randomized manner. The other important aspect is the design of *gossip-based protocol* by which any node, upon deciding the communication partner according to the *gossip* algorithm, decides the content of the message to send to the communication partner. The main contribution of our paper lies in proposing a *gossip-based protocol* using the idea of *random linear coding*, which has previously been used in the context of communication theory for various purposes.

In this paper, we consider a scenario where there are multiple nodes in the network and also multiple messages, but not all messages are with all the nodes to start with. We are interested in designing mechanisms to ensure that all the nodes receive all the messages very fast. We restrict ourselves to *gossip protocols*, so that each node acts based on *local* information, without a centralized controller. Moreover, at each communication instant between nodes, only one *message* (we comment on this aspect later in the paper) can be transmitted. The *gossip algorithm* we consider in this paper is what is popularly known as the *random phone call* model or *rumor mongering* model [4]. In such a model, the system proceeds in rounds. In each round, every node u calls a communicating partner v chosen uniformly at random from from all the nodes. Thus, in the model we consider, the underlying communication graph is complete, in the sense that, each node can potentially communicate with every other node. Alternatively, one can consider a more generic model where a node can only communicate with any one from a given set of neighbors. While the dissemination of a single message was first studied for a system similar to ours, it was later extended to a more general communication model. However, a detailed study with a complete underlying communication graph is important to the understanding of the benefits of out protocol when there are no constraints in the network due to nodes being too far from each other. We propose *gossip-based protocol* using the idea of *random network coding* introduced by Ho et. al. in [7, 8], and compare the protocol with a naive one. The details of the protocols are provided later in the paper.

As we show in the paper, information dissemination schemes based on the concepts of *network coding*, instead of a naive *store and forward* mechanisms, can provide substantial benefits in distributed environments at the cost of a small overhead (if the message sizes are reasonably large) associated with each packet. In networks with fixed communicating graphs, *network coding* can be viewed as a vast generalization of routing where each packet is treated as an algebraic entity that can be operated upon instead of simply storing and forwarding. Essentially, each node in the network sends to each output link any linear function of the data received at the input links. It was shown in [15] that linear network coding can achieve the min-cut bound in networks with multicast connections. There is a significant recent work on network coding [9], especially on the algorithmic aspects of construction of linear network codes [2, 14, 10, 18] for multicast connections. In [7, 8], the authors proposed the novel idea of random network coding. Our

protocol for message dissemination is inspired by this.

The goal of this work is to propose protocols for simultaneous message dissemination. To, understand the main constraints consider Figure 1. There are eight nodes in the network and each of the nodes has

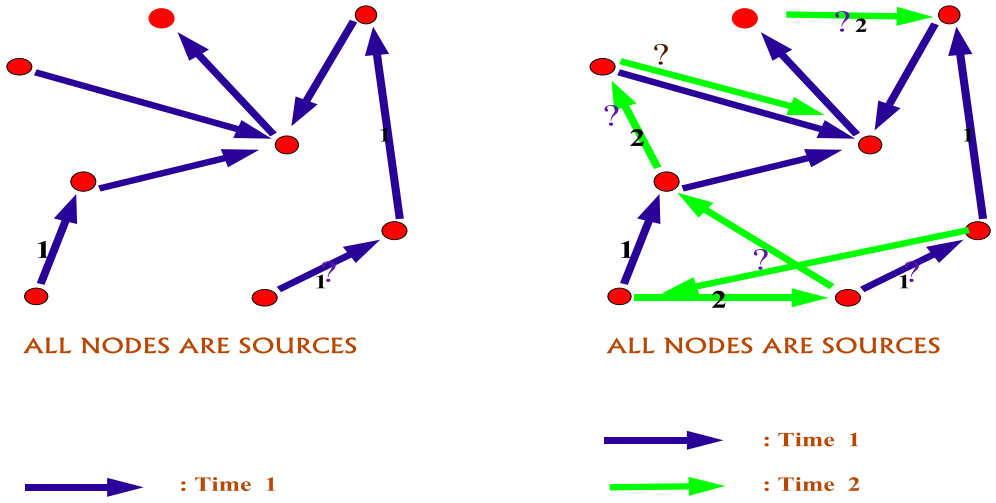

Figure 1: Figure depicting the gossip-based communication in two successive time-steps

a distinct message to start with, and the goal is to disseminate all the messages among all the nodes. In the first time-step, each node picks up a node at random (*gossip* based communication) and transmits the message it has. In the second round, some of the nodes already have two distinct messages. For transmitting a message in the second round, these nodes have to decide which message to transmit without the knowledge of the contents of its communication partner. Obviously, the constraint is imposed by the fact that only one message can be transmitted and nodes only have local knowledge.

We present a *gossip-based protocol* based on random network coding that can simultaneously spread k messages (where each node has only one message initially) among n nodes in time $ck + O(\sqrt{k}\ln(k)\ln(n))$. As the time required to disseminate the messages sequentially is at least $k\log_2(n)$, the dissemination time goes down by a factor of $\Theta(\ln(n))$ in the regime $k \gg (\ln(n))^3$, and by a factor of $\Omega(\sqrt{k}/\ln(k))$ in the regime $k \ll (\ln(n))^2$. Note that any protocol for disseminating the messages simultaneously will require at least k rounds, and so the dissemination time using random network coding in the regime $k \gg (\ln(n))^3$ is order optimal. The key feature in the protocol which helps to attain this bound is an "algebraic mixing" of the messages using *random linear coding*. This is done by viewing each message as an element in a vector space with the scalars in a finite field of appropriate size. We have also shown that a naive uncoded *store and forward* approach of spreading the messages takes $\Omega(n\ln(n))$ time when $k = \alpha n$ for some $\alpha \leq 1$. Note that, for $k = \alpha n$, this is no better than a sequential approach of spreading the messages one after the

other would take a sum-total of $\Theta(n \ln(n))$ (since the time to propagate a single message is $\Theta(\ln(n))$ [11]).

Before going into the details of the protocols and the results, we provide the key intuition behind the power of a coding based approach in the following. Suppose there are k distinct elements in a finite field of size q. Consider two approaches to store the elements in a database with k slots. Suppose each slot chooses an element at random without the knowledge of what the other slots are choosing. Then, the probability that all the elements are there in the database is very small. Now, consider a second approach in which each slot in the database stores a random linear combination of the messages. All the messages can be recovered from the database, only if the linear combination chosen by the slots are linearly independent. Now, there are $((q^k - 1)(q^k - q)(q^k - q^2) \dots (q^k - q^{k-1}))$ ways of choosing k linearly independent combination of the k elements in a finite field of size q. Thus, the probability that the elements can be recovered from the database is much higher in the latter scenario. This is the key idea which is at the heart of the *random linear coding* based protocol we present in this paper.

While the basic details of RLC based dissemination may make it believable that k messages can be disseminated in $\Theta(k)$ rounds, a rigorous derivation poses quite a few technical challenges due to the distributed nature of the system. The proof relies on suitably modeling a time-varying Bernoulli process (where each subsequent toss of a coin has a larger probability of getting a "head") which we have not encountered in the literature. We comment that the main contribution of the paper does not lie in proposing the notion of *random linear coding*, rather in applying the notion to uncoordinated dissemination and demonstrating the gains that can be had by rigorous analysis and simulations. We also provide lower bound on the dissemination time of a *store and forward* protocol called Random Message Selection or RMS. The protocol is a distributed version of the famous *coupon collector* problem where the coupons are distributed in the network. While the derivation of the lower bound relies on the idea behind the *coupon collector* problem, we provide a rigorous derivation as we have not encountered a similar proof in the literature.

The rest of the paper is structured as follows. In Section 2, the model, the protocols considered are described along with a few preliminaries. We state our main analytical results and also report our simulation results in Section 3. In Section 4- 7, we provide detailed analysis of the different protocols considered in this paper. We conclude in Section 8.

2 Model, Protocols, and Preliminaries

2.1 Model and Protocols

Suppose there are n nodes and k messages. Initially, each node has one of the k messages indexed by the elements in the set $M = \{m_1, m_2, \ldots m_k\}$. The nodes are indexed by elements of the set $[n]$.

Assumption 1. *(Initial Condition) Every node has only one out of the k messages initially. If V_{m_i} is the set of nodes that start out with the message $m_i \in M$, we also assume $|V_{m_1}| = |V_{m_2}| = ... = |V_{m_k}|, \forall m \in M$, i.e., each message is equally spread in the network to start with.*

Our theoretical results are derived under this assumption, but we do not think that this assumption is too restrictive as we show in our simulations. Suppose there are only k distinct messages at k nodes initially and no messages with the other $n - k$ nodes. Then there are two phases of message dissemination if all the nodes require all the messages. The first phase ends when every node has at least one message and this might take a time of around $\log_2(n)$. The second phase starts once every node has at least one message and ends with all the nodes having all the messages. Since disseminating k messages takes at least k rounds, for $k \gg \log_2(n)$, the second phase is the dominant phase. Our goal in this paper is to understand the message dissemination time in the second phase (after an initial period of possibly $\log_2(n)$ when all the nodes have some message). Further since, all the messages are identical, Assumption 1 might be viewed as some kind of "average behavior" once every node has some message. We comment that, when k is large, i.e., say $k = \alpha n$ for some $\alpha \leq 1$, the results and the derivations in this paper can be easily extended for the case when some nodes have more than one message and some have none, or when all the messages are there with one particular node to start with. We are interested in obtaining the time required to disseminate all the messages to all the nodes using a rumor mongering approach in the asymptote of large n and k.

Gossip Algorithms:

The system advances in *rounds* indexed by $t \in \mathbb{Z}^+$. The communication graph in round t, G_t, is obtained in a randomized manner as follows. In the beginning of each round, each node $u \in [n]$ calls a communication partner v chosen uniformly from $[n]$. As proposed in [4], we consider two versions of the *rumor mongering* model for message exchange.

Pull: In this model, a message is transmitted from a *called* node to the *caller* node according to a suitable protocol we describe later. Thus, the communication process is initiated by the receiving node.

Push: Here, the message is transmitted from a *caller* node to the *called* node. Thus, the communication process is initiated by the transmitting node.

There can be other variants of these basic models, for instance, by combining *push* and *pull* as considered in [11].

Gossip-Based Protocols:

Having described the model for communication graph in each round, we now describe two protocols or strategies for transmitting a message. The protocols will be adopted by the *caller* node in the *push* model and the *called* node in the *pull* model. Below we describe two protocols for message transmission we consider in this paper.

Random Message Selection (RMS): This is a simple strategy, where the transmitting node simply looks at the messages it has received and picks any of the messages with equal probability to transmit to the receiving node. Thus, if M_v is the set of messages at node v, then v transmits a "random" message e to its communicating partner, where

$$\Pr(e = m) = \frac{I_{(m \in M_v)}}{|M_v|} .$$

In the above $I_{(m \in M_v)}$ is the indicator variable of the event $(m \in M_v)$.

Random Linear Coding (RLC): Suppose the messages are vectors over the finite field $\mathbb{F}_q$ of size $q \geq k$. If the message size is m bits, this can be done by viewing each message as an $r = \lceil m/\log_2(q) \rceil$ dimensional vector over $\mathbb{F}_q$ (instead of viewing each message as a m dimensional vector over the binary field). To this end, let $m_i \in \mathbb{F}_q^r$ (m_i, $i = 1, 2, \ldots k$, are the messages) for some integer r. Thus the messages are over a vector space with the scalars in $\mathbb{F}_q$. All the additions and the multiplications in the following description are assumed to be over $\mathbb{F}_q$. In the RLC protocol, the nodes start collecting several linear combinations of the messages in M. Once there are k independent linear combinations with a node, it can recover all the messages successfully. Let S_v denote the set of all the coded messages (each coded message is a linear combination of the messages in M) with node v at the beginning of a round. More precisely, if $f_l \in S_v$, where $l = 1, 2 \ldots |S_v|$, then $f_l \in \mathbb{F}_q^r$ has the form

$$f_l = \sum_{i=1}^{k} a_{li} m_i, \quad a_{li} \in \mathbb{F}_q,$$

and further the protocol ensures that a_{li}'s are known to v. This can be done with a minimal overhead with each packet in a manner described soon.

Now, if the node v has to transmit a message to u, then v transmits a "random" coded message with payload $e \in \mathbb{F}_q^r$ to u, where

$$e = \sum_{f_l \in S_v} \beta_l f_l, \;\; \beta_l \in \mathbb{F}_q \tag{1}$$

and

$$\Pr(\beta_l = \beta) = \frac{1}{q}, \;\; \forall \beta \in \mathbb{F}_q \; . \tag{2}$$

For decoding purposes, the transmitting nodes also send the "random coding vectors" as overhead with each packet. This can be achieved by padding an additional $k \log_2 q$ bits with each message. To see the precise structure of the overhead associated with a packet, note that the payload part of the transmitted message e in (1) can be represented as follows:

$$\begin{aligned} e &= \sum_{f_l \in S_v} \beta_l f_l \\ &= \sum_{f_l \in S_v} \beta_l \sum_{i=1}^{k} a_{li} m_i \;\; (\text{where} \;\; a_{li} \in \mathbb{F}_q) \\ &= \sum_{i=1}^{k} \theta_i m_i \;\; (\text{where}, \;\; \theta_i = \sum_{f_l \in S_v} \beta_l a_{li} \;\; \in \mathbb{F}_q) \end{aligned}$$

It is the θ_i's that are sent as overhead with the transmitted messages. Thus, once the β_l's are selected in randomized manner according to (2), the transmitting nodes can precisely obtain the values of θ_i's ($i = 1, 2 \ldots k$) and send as overhead. This overhead would clearly require a padding of additional $k \log_2(q)$ bits. We also call the overhead $(\theta_1, \theta_2, \ldots, \theta_k) \in \mathbb{F}_q^k$, the transmitted "code-vector." We simply comment that, if the message size $m \gg \log_2(q)$, then the overhead required with the protocol is minimal. Note that the overload scales with the number of messages being spread simultaneously. The field size q is a design parameter, on which we comment later in Section 3.

The decoding of the messages is not hard to see. In RLC approach, the nodes start collecting the "code vectors" as the system progresses. Once the dimension of the subspace spanned by the received "code vectors" at a node becomes k, then the node can recover all the messages.

We are interested in the finding the expected time (rounds) required for all the nodes to receive (decode) all the messages, and also the time required to receive all the messages with high probability for four cases: RLC with *pull*, RLC with *push*, RMS with *pull*, RMS with *push*.

2.2 Notations and Preliminaries

Our analysis decomposes the system evolution into multiple phases. In many of the cases, we show that the time spent in each phase can be accurately described by a random variable with *discrete phase type* distribution. Below we provide the definition of such a random variable. The definition is presented in a manner to serve our purposes.

Definition 1. *(Discrete phase type distribution)*

Consider a discrete time Markov Chain X_k on the state space $\mathcal{A} \cup \mathbb{Z}^+$, where $\mathcal{A}$ is the absorption state. Also suppose $X_0 = 1$ and the transition probability matrix P has a structure such that $p_{s,r} = 0$ if $r < s$ or $r > s + 1$ for $r, s \in \mathbb{Z}^+$. Further suppose $P_{s,\mathcal{A}} > 0$ and $\Pr(\lim_{k\to\infty} X_k = \mathcal{A}) = 1$. Then, the random variable T, defined by

$$T = \inf\{k : X_k = \mathcal{A}\}$$

has a discrete phase type distribution with parameters given by the transition probability matrix of the underlying Markov chain. We call T the absorption time of the corresponding Markov chain.

For the Markov chain shown in Figure 5 (the given Markov Chain is useful to some of our later results and so the figure is pushed to a later page), the time to reach the state $\mathcal{A}$ starting at state 1 at time zero, has a *discrete phase type* distribution. Note that, a geometric random variable is a special case of the *discrete phase type* distribution with $p_{s,\mathcal{A}} = 1 - p_{s,s+1} = p > 0$.

We use the notation $X \prec_{st} Y$ for two random variables to imply that X is smaller than Y in a stochastic ordering sense [17], i.e., $\Pr(X \geq a) \leq \Pr(Y \geq a)\ \forall\ a$. Sufficient conditions for $X \prec_{st} Y$ to hold are provided in [17]. We also use the notation $X \sim Y$ to imply that the distribution of the random variables X and Y are identical. Further, we use the standard abbreviations *w.p.* and *a.s.* to mean "with probability" and "almost surely", respectively.

Further, we use the notation $\text{Geom}(p)$ for a geometric random variable with parameter p, i.e., $\Pr(\text{Geom}(p) = k) = (1-p)^{k-1}p$ for $k \geq 1$. We also use the notation $\text{Bin}(N, p)$ for a Binomial random variable with parameters N and p, i.e., number of heads in N tosses with a coin with probability of "head" p.

We comment that the underlying probability space has a probability measure determined by the random communication graphs in each round and the random transmitted messages. By a natural abuse of terminology, in our analysis and discussion of the RLC protocol, we also refer to "dimension of the subspace spanned by the code-vectors received by the node" as "dimension of a node." Throughout this paper, we also use the terms "round" and "time" interchangeably.

2.2.1 A useful result on the RLC protocol

We now state and prove a useful, but simple and intuitive result which is key to demonstrating the benefits of the RLC protocol. In the following, we assume that a coded message is transmitted from node v to node u. It is implicit that, with a "pull" mechanism u is the caller node, and with a "push" mechanism v is the caller node.

Lemma 2.1. *Suppose node v transmits a coded message to node w in a particular round using the RLC protocol. Let S_u^-, S_w^- and S_v^- denote the subspaces spanned by the code-vectors with u, w, and v respectively at the beginning of the round. Let S_w^+ denote the subspace spanned by w at the end of the round, i.e., after receiving a coded message from v according to the scheme described by the RLC protocol. Then,*

$$a)\ \Pr(S_w^+ \not\subseteq S_u^- | S_w^- \subseteq S_u^- \ , S_v^- \not\subseteq S_u^-) \geq 1 - \frac{1}{q}$$

$$b)\ \Pr(\dim(S_w^+)\dim(S_w^-) | S_v^- \not\subseteq S_w^-) \geq 1 - \frac{1}{q}\ ,$$

where q is the size of the field.

Proof. The result is reminiscent of the Schwartz-Zippel lemma. However, the framework here is slightly different. The result follows almost immediately from the fact that, if $S_v^- \not\subseteq S_u^-$, then S_v^- must have a component orthogonal to S_u^-. We make this observation precise in the following. We will simply prove the first part.

Consider the event $S_v^-, \not\subseteq S_u^-$, conditioned on which we want to calculate the probability. Clearly, there exists $\{g_1, g_2, \ldots, g_l\} \subseteq S_v^-$ ($g_i \in \mathbb{F}_q^k$) such that each of the g_i's have a component orthogonal to the subspace S_u^-. Let

$$g_j = v_j + u_j\ ,$$

where $0 \neq v_j \perp S_u^-$ (and hence $v_j \perp S_w^-$) and $u_j \in S_u^-$. Suppose, the called node v decides to send a coded message to the caller node u in which g_j is multiplied by a random element β_j. Clearly,

$$\sum_{j=1}^{l} \beta_j v_j \neq 0 \ \ \Rightarrow S_w^+ \not\subseteq S_u^- \Rightarrow \dim(S_w^+) > \dim(S_w^-)\ . \tag{3}$$

Since $v_i \in \mathbb{F}_q^k$, we can represent the v_i's as $[v_1, v_2, \ldots v_l]^t = A$ where A is an $(l \times k)$ matrix in $\mathbb{F}_q$. Then $\sum_{j=1}^{l} \beta_j v_j = 0$ iff

$$[\beta_1, \beta_2, \ldots, \beta_l]A = 0\ .$$

The preceding set of equations in β_j's has k equations and l variables and so it has at most q^{l-1} solutions in $\mathbb{F}_q$. Further since the β_j's are chosen at random,

$$\Pr(\sum_{j=1}^{l} \beta_j v_j = 0) \leq \frac{q^{l-1}}{q^l} = \frac{1}{q}$$

from which the result follows owing to (3). □

3 Main Results of the Paper

We now describe the main results of the paper. The detailed analysis of each of the protocols leading to these results are provided in the subsequent sections. The stated results will also be stated in a more detailed form in the subsequent sections.

Our first result is on the performance of RLC with *pull* mechanism.

Theorem 3.1. *Suppose $q \geq k$. Let $\overline{T}_{RLC}^{pull}$ be the random variable denoting the time required by all the nodes to get all the messages using an RLC approach with pull mechanism. Then, under Assumption 1,*

$$\overline{T}_{RLC}^{pull} \leq 3.46k + O(\sqrt{k\ln(k)}\ln(n)), \;\; w.p.\; 1 - O(\tfrac{1}{n})$$

Further, if T_{RLC}^{pull} is the time required for a particular node to get all the messages, then

$$\mathbb{E}[T_{RLC}^{pull}] \leq 3.46k + O(\sqrt{k\ln(k)}\ln(n))\ .$$

We also have a similar result with a *push* based mechanism.

Theorem 3.2. *Suppose $q \geq \max(k, \ln(n))$. Let $\overline{T}_{RLC}^{push}$ be the random variable denoting the time required for all the nodes to get all the messages using an RLC protocol with push mechanism. Then, under Assumption 1,*

$$\overline{T}_{RLC}^{push} \leq 5.96k + O(\sqrt{k}\ln(k)\ln(n)), \;\; w.p.\; 1 - O(\tfrac{1}{n})$$

If T_{RLC}^{push} is the time required for a particular node to get all the messages with RLC based push, then

$$\mathbb{E}[T_{RLC}^{push}] \leq \; 5.96k + O(\sqrt{k}\ln(k)\ln(n)),$$

Remark 1. *The following extensions of the results are routine.*

1. *Suppose $k = \alpha n$ and $\alpha < 1$ is a fixed constant. If there is only one copy of each of the k messages with k different nodes initially, so that there are some nodes with no messages, then one can show that the dissemination time is asymptotically at most ck for a suitable constant $c > 0$.*

2. *Suppose $k = \alpha n$ for some fixed $\alpha < 1$, and suppose that initially there are k distinct messages at a single node. Then, similar to Theorem 3.1 and 3.2, one can show that the dissemination time is asymptotically at most $c'k$ for a suitable constant $c' > 0$.*

3. *If we simply restrict q to $q \geq k$ in Theorem 3.2, we can show that the dissemination time is less than $ck + O(\sqrt{k}\ln(k)\ln(n))$ for a suitable c (larger than 5.96).*

The next natural question is, what if the nodes do not manipulate the packets and simply *store* and *forward* the packets? We show that, in one such protocol as we have described in the paper, which we call RMS or "Random Message Selection," one can do no better than the case when the messages are disseminated in the network sequentially one after the other. While the protocol is simple, as we have not encountered any careful stochastic analysis of such a protocol in the literature, we state and prove lower bound based results of such a protocol. We use the notation $T = \Omega(n\ln(n))$ to imply $T \geq cn\ln(n)$ for a suitable constant $c > 0$.

Theorem 3.3. *Suppose $k = \alpha n$ for some $\alpha \leq 1$, and let $T_k^{RMSpull}$ be the time required for all the nodes to get all the k messages using an RMS protocol with pull mechanism. Then, we have*

$$\mathbb{E}T_k^{RMSpull} = \Omega(n\ln n)$$

and

$$\lim_{k\to\infty} \Pr\left(T_k^{RMSpull} = \Omega(n\ln(n))\right) = 1$$

We also have a very similar result for RMS with a *push* based mechanism.

Theorem 3.4. *Let $k = \alpha n$ for some $\alpha \leq 1$, and let $T_k^{RMSpush}$ be the time required for all the nodes to get all the k messages using an RMS protocol with push mechanism. Then, we have*

$$\mathbb{E}T_k^{RMSpush} = \Omega(n\ln(n))$$

and

$$\lim_{k\to\infty} \Pr\left(T_k^{RMSpush} = \Omega(n\ln(n))\right) = 1$$

A few comments are in order which we note below:

1. In gossip-based communication with one message, it takes $\Theta(\ln(n))$ time for complete dissemination to occur with high probability. Thus, if the k messages are disseminated sequentially one after the other, it will take $\Theta(k\ln(n))$ time to disseminate all the messages. According to our results, for

$k \gg (\ln(n))^3$, the time to disseminate all the messages using RLC is asymptotically at most, $3.46k$ with *pull* and $5.96k$ using *push*. Thus, an RLC based dissemination can provide substantial gains (reduction in dissemination time by a factor of $\Theta(\ln(n))$) in message dissemination time when k is larger than $(\ln(n))^3$. Depending upon the size of the network, the gain in dissemination time, which is $\Theta(\ln(n))$, can be quite large. Further, since no protocol can disseminate the messages in time less thn k, the asymptotic dissemination time of ck is order optimal. For smaller values of k when $k \ll (\ln(n))^2$, the reduction in dissemination using RLC is at least a factor of $c_1\sqrt{k}/\ln(k)$ with push and a factor of $c_2\sqrt{k/\ln(k)}$ with pull for suitable c_1 and c_2.

2. The notion of "rounds" or "discrete-time step" has to be interpreted suitably. One round of message transfer from one node to another simply refers to one message transferred from one node to the other. One might also consider using parallel channels between the nodes, in which case, there might be multiple messages exchanged between two nodes in one time-step. We believe our results can be suitably modified in such a scenario.

3. Note that, if there is no bandwidth constraint (i.e., if a transmitting node can transmit its entire database) between two communicating nodes, the dissemination time is simply $\Theta(\ln(n))$ for any k. This is since the system behaves as if there is only one message for which the dissemination time is $\Theta(\ln(n))$ [11].

4. An interesting quantity is the total amount of information that is exchanged. Consider the regime $k \gg (\ln(n))^3$. If each message is of size m bits, the total amount of information exchanged in the RLC protocol with $q \approx k$ is less than $cnk(m + k\log_2(k))$ for some constant $c > 0$. In the case of sequential dissemination (where messages are disseminated one after the other), this quantity is $c_1 nk\ln(n)(m + \log_2(k))$ (additional $\log_2(k)$ bits for identifying each message) for some constant $c_1 > 0$. Further, any protocol will require at least nkm bits of transmission. Note that since m is in bits, the additional overhead with RLC is roughly $100(k\log_2(k)/m)\%$, which is typically a small quantity. For example, with $k = 100$, the overhead is 1% for $m = 100$ KB and it is 0.1% for $m = 1$ MB. We simply note that the overhead does not grow with the size of the messages or available bandwidth and simply depends on the number of messages that are to be disseminated simultaneously. In any case, RLC is useful only when, $k\log_2(k) \ll$ (size of each message in bits).

5. We would also like to point out that the main computational aspect of RLC is in the end of the dissemination time and it takes $O(k^3)$ operations. This is typically not large for $k \leq 1000$ and

with modern processors. The computation involved for each message transmission is no more than $O(mk/(\log_2(q))$ operations, where m is the size of a message. While this computation time is not much for typical values of parameters and modern processors, whether this computation time is large or small depends on the values of m, k and the processor [1]. Our goal of this paper is more fundamental, to show that RLC based message dissemination achieves the optimal dissemination time (in an order sense) without requiring to exchange the list of messages.

The power of a coding based approach comes from the fact that packets are treated as algebraic entities which can be operated upon.

Remark 2. *The inherent advantage of RLC comes from "coding." The RMS scheme cannot do as well even if packets were chopped up into multiple parts, or multiple packets were combined into large packets. To see this, suppose each packet of size m is chopped into r mini-packets of size m/r each. There are kr packets in the system. Suppose, there are r mini-rounds within each round for the transmission of these min-packets. The new RMS scheme will take $\Omega(kr\ln(kr))$ mini-rounds or equivalently time worth $\Omega(k\ln(kr))$ rounds in the original scheme. Also, a very similar modification for RMS scheme can be done with combining a fixed number of packets. Hence, splitting or combining packets cannot help the non-coding nature of RMS scheme to achieve the optimal order attained by a coding based scheme.*

The RLC scheme does perform better than the RMS scheme, but, we did not allow any overhead in the RMS protocol. However, even if a minimal overhead is allowed in the RMS protocol, the protocol cannot benefit from any possible extra information.

3.1 Simulation results

3.1.1 Comparison of RLC with RMS and sequential dissemination

In this section, we provide some simulation results with *push* based dissemination mechanism. The purpose of the simulations is two-fold. First, we want to get a more accurate idea about the dissemination time (the theoretical results simply provide an upper bound). Secondly, we wish to investigate if there are gains to be had by using RLC for very small values of k.

In all our simulations, there are k nodes that start with k distinct messages and all the other $n-k$ nodes do not have any messages to start with (Note that the theoretical upper bounds are derived assuming every node has some message initially as noted in Assumption 1). We also choose $q = k$ in all the cases.

[1] for example, some back of the envelope calculations for $m = 1$ MB, $k = 100$, and a 1 GHz processor yields that this computation time is of the order of a tenth of a millisecond

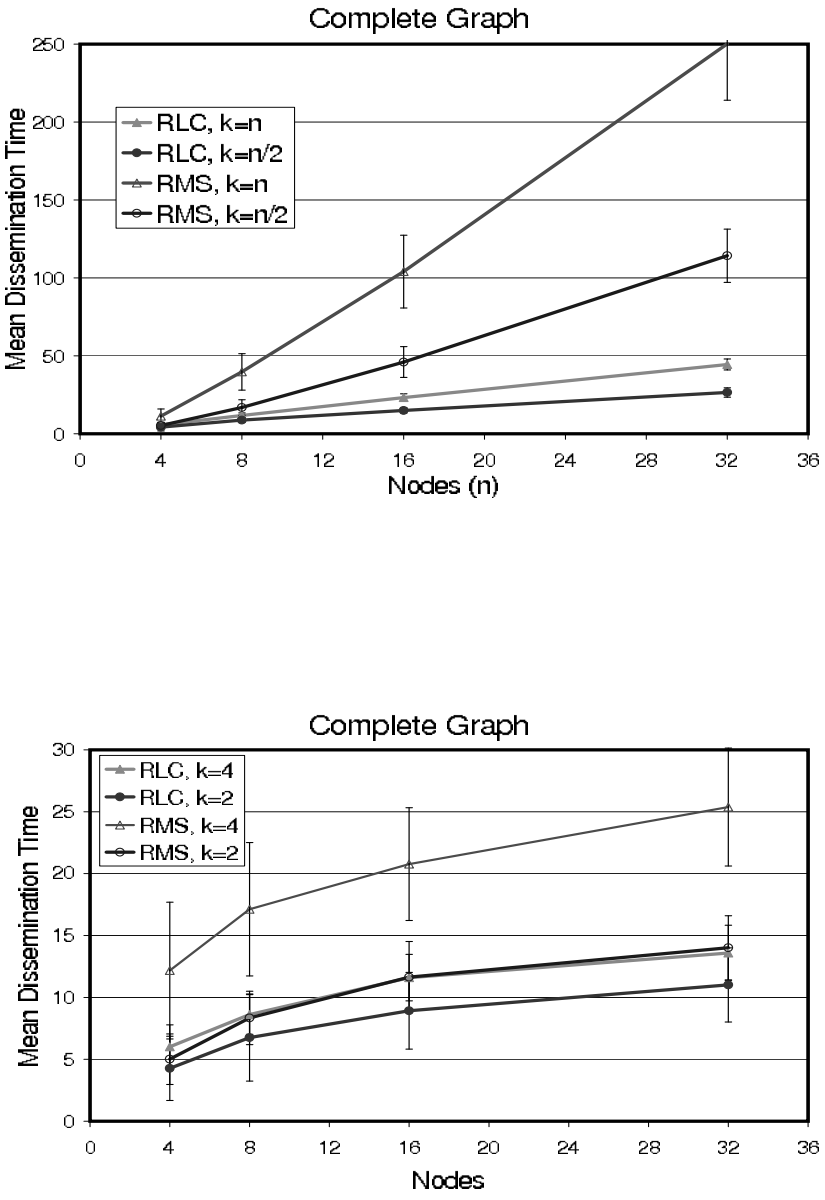

Figure 2: Plots showing the dissemination time with RLC and RMS protocol for different values of n (number of nodes) and k (number of messages). The underlying communication graph is complete, i.e., every node can pick any other node as its communication partner.

In Figure 2, we show the mean complete dissemination time with RLC and RMS protocol. The mean is obtained by averaging the complete dissemination time (the time by which all the nodes get everything) over 100 runs. In the plots on the top panel of Figure 2, we show the how the dissemination time varies with the number of nodes n, when the number of messages is $k = n$ and $k = n/2$. The RLC protocol for message dissemination far outperforms the RMS protocol. The RMS protocol also seems to perform identically to sequential dissemination of the messages. In the bottom panel, we have also shown the plots when number of messages k is fixed at two and four. The purpose of this paper is to explore whether a random linear coding based protocol can be useful in disseminating messages simultaneously. Thus, the important question is not whether RLC outperforms RMS or not, rather, whether simultaneous dissemination of the messages can expedite the dissemination process or not. Consider the dissemination time with $n = k = 32$. The mean dissemination time is around 45 rounds. As it is well known that disseminating a single message takes around $\log_2(n) + \log_2 \log_2(n) \approx 7$ rounds [11], disseminating $k = 32$ messages would take around 224 rounds if the messages are disseminated one after the other. Thus, simultaneous dissemination of messages using RLC protocol reduces the time to less than one fourth (also note that the RMS protocol does no better than sequential dissemination). A similar trend can be observed in the case $n = 32$ and $k = 4$. In this case, the mean dissemination time of RLC protocol is around 13 rounds, whereas, disseminating the messages one after the other would take around 28 rounds. Clearly, RLC protocol can provide appreciable gains in dissemination time even for small number of messages. We again remind the reader that, RLC protocol comes with a little overhead of $k \log_2(k)$ additional bits per transmission, which, in all of the cases considered in the simulations (i.e.,$k \leq 32$), is at most 20 bytes. For most applications, the size of a message is likely to be much larger than this. It appears based on some of the simulations we have done that, the mean time to disseminate k messages is close to $1.5k + \log_2(n)$ when k nodes start with k messages and other $n - k$ do not have any messages to start with. Thus, the upper bound in the theoretical results are overestimates.

Why does the RMS protocol perform badly and RLC do well? In RMS, since messages are picked at random, more the messages are at a node, the more likely it is that the received message is already found at the node. RLC protocol overcomes this in the following way. In this the nodes build up dimension of sub-space spanned by the received code-vectors. By Lemma 2.1, the probability that the dimension increases due to a newly received coded message does not go down as the dimension gets closer to k or full-rank. In Figure 3, we show plots for the time taken for the dimension of the various nodes to increase to different values. We show plots for nodes that take longest to receive the messages, nodes that take least amount of time, and also for a typical node.

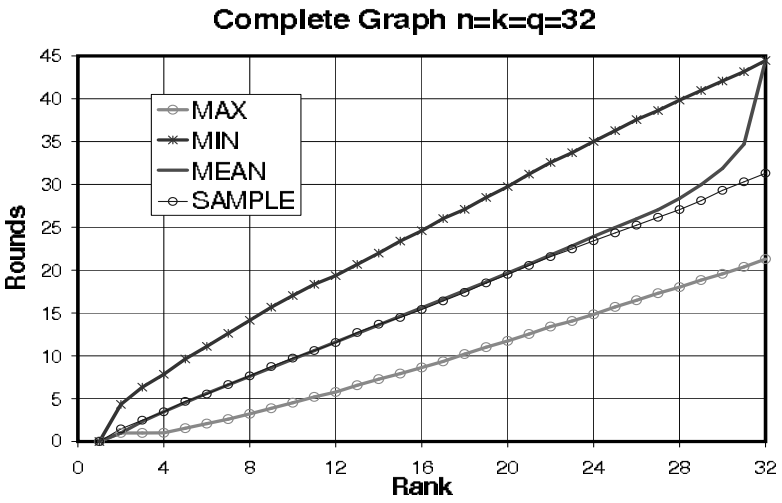

Figure 3: Plot showing how the rank builds up in the RLC protocol

3.1.2 Comparison of RLC with a modified version of RMS

In this paper we demonstrate that, RLC based message dissemination can provide substantial gains over a random message selection (RMS) based scheme or sequential dissemination. One natural comarison of RLC based dissemination can be with the following modified version of RMS where node u, before transmitting messages to v, seeks the list of messages v has, and then picks a message randomly from the ones u has but v does not have. This modified version of RMS would require an extra round per message exchange for the nodes to exchange the list of messages. In Figure 4, we compare RLC based dissemination with the modified RMS. By taking into account the extra round required by this modified RMS, we can see that the average dissemination time of an RLC based dissemination is less by a factor of two. In other words, the results indicate that an RLC based dissemination achieves the exact effect of exchanging the list of messages without having to do so!

Remark: The modified version of RMS described in this subsection is similar to the one used by bit-torrent file sharing system. The analysis in this paper with RLC does not carry over directly to a bit-torrent like message dissemination for the setting in this paper. In this paper, we have focused on disseminating multiple messages among all the nodes. The problem is mostly motivated by earlier studies on gossip-based message dissemination with one message, where the inherent assumption is that messages are disseminated

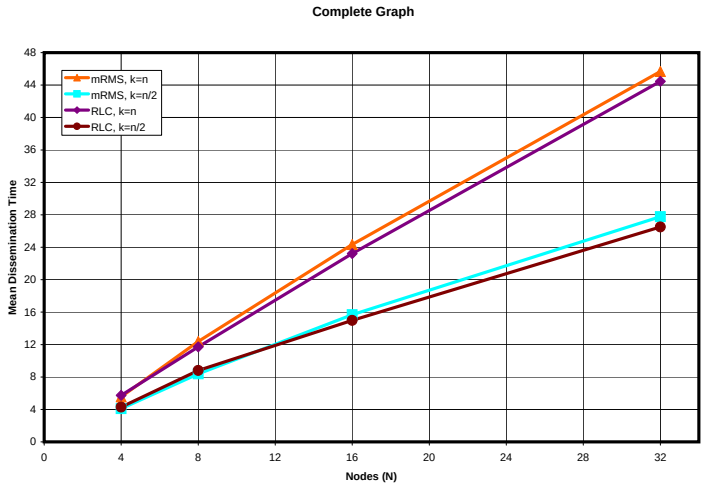

Figure 4: Plots showing the dissemination time with RLC and a modified RMS protocol (where a transmitting node randomly picks a message from the disjoint set of messages) for different values of n (number of nodes) and k (number of messages). The underlying communication graph is complete, i.e., every node can pick any other node as its communication partner.

sequentially. However, we would like to bring in attention the conference paper [1], where we have analyzed a bit-torrent like RMS (the one pointed by the reviewer) with RLC and also with traditional erasure codes for a disributed file storage system. The setting there is somewhat different as certain nodes wish to collect all the pieces of a file and other nodes simply act as limited storage elements and are not interested in gathering pieces of the file. The analysis in [1] show a clear advantage of an RLC based storage over bit-torrent like RMS, namely, the fact that, the probability that a certain fraction, say x of download is completed after contacting, say r storage elements, is siginificantly more than that of bit-torrent like RMS (the paper also contains the analysis for a Reed-Solomon code like storage). Furthermore, RLC can also be used to provide security (the details are in a submitted longer version of [1]).

3.2 Key idea behind the results using a mean-field approach

Before we proceed to analyze the protocols in detail, we provide an intuition behind our results, and also comment on the analysis approach of the protocols. The argument in this subsection is not rigorous and far

from formal, and is only to provide a heuristic behind the optimal order attained by RLC mechanism. In the subsequent sections, we formally prove the results.

First consider the RMS protocol and let us concentrate on any particular node, u. Since u starts with one message at round zero, in the initial rounds, any communication from some other node is very likely to provide u with a new message. However, as u gathers more and more messages, any new message is more and more likely to be something u already has (recall the famous *coupon collector* problem [5]). Indeed, our proof of the result with RMS protocol shows that, the system takes $\Omega(k \ln(k))$ rounds just to receive the last $k/2$ messages. Thus, the performance of the RMS protocol deteriorates once a node already has roughly half the total messages.

Now consider the RLC protocol with *push* mechanism (a similar intuition can be given for the *pull* model). As before concentrate on a particular node u. The node u keeps receiving *code vectors* and decodes all the messages once the dimension of node u is k. Suppose the dimension of node u is i. We are interested in finding an expression for the number of rounds for which u has dimension i. First, lets classify the nodes as "helpful" and "unhelpful" as follows. We call a node "helpful" to u, if the subspace spanned by its *code vectors* does not lie in that of u. Otherwise, a node is "unhelpful" to u. The first point to note is that, if u is pushed by a helpful node, the conditional probability of node u increasing its dimension to $i+1$ is at least $1 - 1/q$ by Lemma 2.1. This is true for any unhelpful node as well, i.e., if any node that is unhelpful to u is pushed by a node that is helpful to u, the unhelpful node increases its dimension (and also becomes a helpful node, provided u has yet to increase its dimension) with probability at least $1 - 1/q$. Let ϕ be the fraction of helpful nodes when node u has dimension i for the first time. It is not hard to argue that $\phi \geq (k-i)/k$, with equality corresponding to the case when node u has recovered i messages. This is because, if u has recovered i messages, then there is at least the $(k-i)/k$ fraction of nodes that started with the remaining $k-i$ messages. Let us divide the time spent by u at dimension i into two phases. The first phase ends when ϕ becomes at least $1/2$. The second phase starts at this point and ends when u increases its dimension. This total time in the two phases will clearly give an upper bound on the time the node u spends at dimension i, as it is possible that the dimension of u icreases from i before the first phase ends. Consider the first phase that ends when the fraction of helpful nodes exceeds $1/2$. Now, the number of unhelpful nodes pushed by a helpful node is roughly proportional to the number of helpful nodes, which is $n\phi$ to start with (i.e., first time u has dimension i). To see this, let there be r unhelpful nodes and $(n-r-1)$ helpful nodes, where $r/n > 1/2$. Each unhelpful node is pushed by at least one helpful node with probability $1-(1-1/n)^{n-r-1}$ which is greater than $(n-r-1)/(2n)$. Thus, on an average the number of unhelpful node pushed by a helpful node is greater than $r(n-r-1)/(2n)$ which is greater than $(n-r-1)/4$ for $r/n > 1/2$. Thus, on

average, roughly $n\phi(1-1/q)/4$ unhelpful nodes become helpful nodes after one more round of message exchange. Thus, we have after one additional round of message exchange

$$\phi \leftarrow \phi + \phi(1-1/q)/4 = \phi(5/4 - 4/q) \; .$$

It follows that, the updated value of ϕ after an additional round of message exchange satisfies $\phi \geq \frac{k-i}{k}(5/4 - 4/q)$. Let us suppose $q > 16$. Thus, after r rounds of message exchanges ϕ becomes at least $((k-i)/k)(5/4-4/q)^r$. A simple calculation shows that, after roughly $\ln(k/(2(k-i))/\ln(5/4-4/q)$ rounds (this is the an upper bound on the length of the first phase), the fraction of helpful node becomes at least $1/2$. At this point, the second phase starts with $\phi > 1/2$. However, any helpful node can increase the dimension of u with probability at least $(1/n)(1-1/q)$, since any helpful node communicates with u with probability $1/n$ and increases the dimension with probability $1-1/q$ at least. Since there are at least $n/2$ helpful nodes now, the probability that u does not increase its dimension is at most $(1-\frac{1}{n}(1-\frac{1}{q}))^{\frac{n}{2}} \approx \frac{1}{\sqrt{e}}$ (for large n). Thus, once there are at least $n/2$ helpful nodes, the mean time for u to increase its dimension is $1/(1-1/\sqrt{e})$ which is the length of the second phase. Thus, on an average, the total time T_i that u spends while it has dimension i is no more than the sum of the time it takes until there are $n/2$ helpful nodes, and $1/(1-1/\sqrt{e})$. Thus,

$$T_i \leq \frac{\ln(k/(2(k-i))}{\ln(5/4-4/q)} + \frac{1}{(1-1/\sqrt{e})}$$

which implies

$$\begin{aligned}
\sum_{i=1}^{k-1} T_i &\leq \sum_{i=1}^{k-1} \frac{\ln(k/(2(k-i))}{\ln(5/4-4/q)} + (k-1)\frac{\sqrt{e}}{\sqrt{e}-1} \\
&= \frac{\ln(k^{k-1}/(2^{k-1}(k-1)!)}{\ln(5/4-4/q)} + (k-1)\frac{\sqrt{e}}{\sqrt{e}-1} \\
&= O(k)
\end{aligned}$$

Thus, a mean-field argument indicates that RLC with *push* attains the optimal order. However, the preceding argument is far from rigorous and a rigorous analysis requires careful analysis of various stochastic processes.

An almost similar heuristic can be provided for *pull*. The only difference is that, here we keep track of the unhelpful nodes. More precisely, starting with the fraction of unhelpful nodes $(1-\phi)$, after one more round of message exchange, the fraction of unhelpful nodes becomes at most$(1-\phi)^2 + (1-\phi)\phi(1/q)$. The first term accounts for the event that an unhelpful node stays so if it pulls from another unhelpful node, and the second term accounts for the event that even if an unhelpful node pulls from a helpful node, with

probability at most $1/q$ (Lemma 2.1) it may not increase its dimension. Using this, we can find the time after which there are at most $n/2$ unhelpful nodes.

We end this section with a few words on the proofs. Intuitively, for the first $k/2$ dimensions (or $k/2$ messages with RMS) any communication is likely to be helpful, with or without coding (RLC or RMS). However, we show that, the benefits of a coding based approach remains until the dimension is almost k, more precisely, until the dimension is $k - \Theta(\sqrt{k\ln(k)})$ using *pull*, and $k - \Theta(\sqrt{k}\ln(k))$ using *push*. We show that it takes $O(n)$ time for the dimension of a node to reach, $k - \Theta(\sqrt{k\ln(k)})$ using *pull*, and $k - \Theta(\sqrt{k}\ln(k))$ using *push*. However, the time to increase the dimension by one cannot be worse than the time to receive a message with a single message based dissemination which takes $\ln(n)$ rounds. Thus, increasing the dimension from $k - \Theta(\sqrt{k}\ln(k))$ to k will take $O(\sqrt{n}(\ln(n)^2)$ time in the worst case. Thus its takes $O(n)$ time to decode all the messages.

In light of the above discussion, we decompose our analysis of RLC protocol into three regimes: when the dimension of a node is less than $k/2$, when the dimension of a node lies between $k/2$ and $k - \Theta(\sqrt{k\ln(k)})$ (in a *pull* based approach), and when the dimension of a node is larger than $k - \Theta(\sqrt{k\ln(k)})$ (with *pull*). In the case of *push*, the term $k - \Theta(\sqrt{k\ln(k)})$ replaced by $k - \Theta(\sqrt{k}\ln(k))$.

We also analyze the RMS protocol which is a distributed version of the coupon collector problem. In the coupon collector problem, there are k coupons and the coupons are drawn uniformly at random with replacement until each coupon is drawn at least once. In our setting, by viewing each message as a coupon of distinct kind, each node tries to collect the coupons by choosing a node at random in each round. Intuitively, we are not likely to improve upon the number of drawings by distributing the coupons, and so the number of rounds in RMS protocol until a node gets all the messages is at least $k\ln(k)$ which is the expected number of drawings in the coupon collector problem. We formally show that, this is indeed the case, i.e. the RMS does no better than the coupon collector problem.

The details of the analysis with the all the protocols are provided in the next few sections.

4 Random Linear Coding with "Pull"

Our first observation is that, once the dimension of the subspace spanned by the coded messages received by a particular node becomes k, that node can recover all the messages successfully. We divide the time required for any node to decode all the messages into $k-1$ phases, where the index of a phase represents the the dimension of the node. Thus, in the i^{th} phase, the subspace spanned by the code-vectors received by a node has dimension i.

Let T_i be the random variable denoting the time spent in phase i by a typical node.

4.1 Stochastic bounds

We show that the random variable T_i can be upper bounded by an appropriate random variable with *discrete phase type* distribution in a stochastic ordering sense.

To characterize the random variables T_i, we break up our analysis into three cases depending on whether $i \leq k/2$, $k/2 < i < k - 14\sqrt{k\ln(k)}$, or $i \geq k - 14\sqrt{k\ln(k)}$. As explained before, when $i \leq k/2$, every transmission is likely to be helpful to a node in any case, and so coding does not provide substantial benefit. In the case, $k/2 < i < k - 14\sqrt{k\ln(k)}$, the benefits of the coding based approach show up. However, when $k/2 < i < k - 14\sqrt{k\ln(k)}$, there may not be any additional benefits from coding, but, even under a worst case assumption, it does not take much time to decode all the messages once in this regime.

For the case $i \leq k/2$, we have a trivial stochastic upper bound on T_i from Lemma 2.1.

Lemma 4.1. *For $i \leq k/2$,*

$$T_i \prec_{st} Y_i \ ,$$

where,

$$Y_i \sim \textit{Geom}\left(\tfrac{1}{2}(1 - \tfrac{1}{q})\right) \ .$$

Proof. Let $S_u(t)$ denote the subspace spanned by the code-vectors with node u at the beginning of round t. Suppose that $\dim(S_u(t)) = i$. Also, let v be the node called by u in this round. The subspace spanned by the code-vectors with node u at the end of the communication in round t is same as that in the beginning of round $t+1$ which we denote by $S_u(t+1)$.

At the beginning of round t, let $E_u(t)$ denote the nodes that cannot help u if called by u, i.e.,

$$E_u(t) = \{v : S_v(t) \subseteq S_u(t)\}$$

We also denote the fraction of such nodes by $h_u(t) \stackrel{\triangle}{=} |E_u(t)|/n$. Note that, by Lemma 2.1, we have, $h_u(t) \leq i/k$. To see this, let I_j be the set of nodes that start with the message m_j at time zero. Clearly, $\exists(I_{l_1}, I_{l_2}, \ldots I_{l_{k-i}})$ such that $E_u(t) \cap I_{l_j} = \emptyset$, $\forall j \leq k - i$. Otherwise, it follows from the definition of $E_u(t)$ that the node u will have access to at least $i+1$ messages. Thus $E_u(t) \subseteq {E'}_u(t)$ where

$${E'}_u(t) = [n] \setminus \bigcup_{j=1}^{k-i} I_{l_j}$$

from which it follows that $|E_u(t)| \leq |E'_u(t)| = ni/k$ since $|I_j| = n/k \ \forall j$. Since u calls any node outside $E_u(t)$ with probability $1 - |E_u(t)|/n$, it immediately follows using Lemma 2.1 that

$$\begin{aligned}\Pr(\dim(S_u(t+1)) > \dim(S_u(t))) &\geq (1 - \tfrac{|E_u(t)|}{n})(1 - \tfrac{1}{q}) \\ &\geq (1 - \tfrac{i}{k})\left(1 - \tfrac{1}{q}\right) \\ &\geq \frac{1}{2}\left(1 - \frac{1}{q}\right)\end{aligned}$$

and hence the result. □

The above calculation does not rely actually on the use of *random coding* which is the essence of RLC protocol, and also does not work when i is close to k. We show that, in the regime $k/2 < i < k - 14\sqrt{k\ln(k)}$, the advantage of a *random coding* based approach shows up very clearly.

Lemma 4.2. *Suppose $q \geq k$. Then, for $k/2 \leq i \leq k - 14\sqrt{k\ln(k)}$,*

$$T_i \prec_{st} Y_i \ ,$$

where, Y_i is a discrete phase type distribution on the finite state space $\{1, 2, 3, \ldots \overline{s}_i\}$ with transition probability structure given by

$$p_{s,s+1} = (1 - \tfrac{1}{n^3})(1 - p_{s,\mathcal{A}}), \ \ s \leq \overline{s}_i \tag{4}$$

$$p_{s,\mathcal{A}} = (1 - \overline{h}_s)(1 - \tfrac{1}{q}), \ \ s \leq \overline{s}_i \tag{5}$$

$$p_{\overline{s}_i,\mathcal{A}} = 1 - p_{\overline{s}_i,\overline{s}_i} = \tfrac{1}{2}(1 - \tfrac{1}{q}) \tag{6}$$

where

$$\overline{h}_s = (\tfrac{i}{k})^{2^s}\left((1 + \tfrac{1}{q})(1 + 7\sqrt{\tfrac{\ln(n)}{n}})\right)^{2^s - 1} \tag{7}$$

and

$$\overline{s}_i = \min\{s : \overline{h}_s \leq \tfrac{1}{2}\} \ .$$

The Markov Chain corresponding to the discrete phase type distribution of Y_i is shown in Figure 5.

Proof. Let t be the first time (round) at which the node u has dimension i. Let $t + T_i$ be the first time when the dimension of of the node u increases from i.

Let $S_v(s)$ be the subspace spanned by the code-vectors at node v in the beginning of round s. At the beginning of round s, let $E_u(s)$ be the nodes which cannot help u if called by u in round s, i.e.,

$$E_u(s) = \{v : S_v(s) \subseteq S_u(s)\}$$

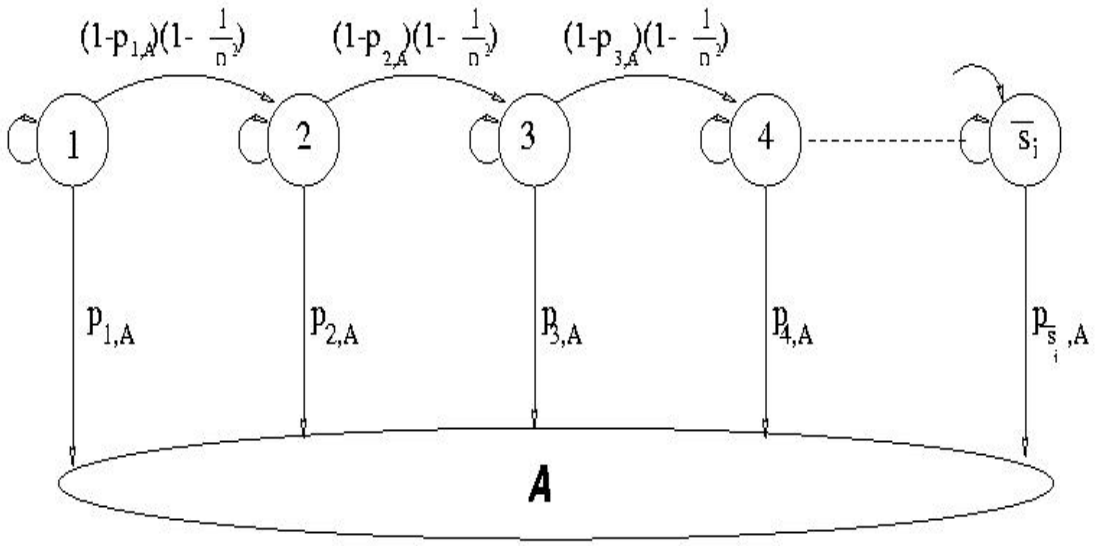

Figure 5: The first time to reach the state $\mathcal{A}$ in the Markov Chain gives an upper bound (in a stochastic order sense) on the time spent by a node while the dimension is i. The values of $p_{s,\mathcal{A}}$ are given by (4)-(6) in Lemma 4.2, and given by (11)-(13) in Lemma 5.3.

We also denote the fraction of such nodes by $h_u(s)$ so that $h_u(s) = |E_u(s)|/n$. We denote by $p_u(s)$ the probability that the dimension of node u remains i after one round of message exchange in the s^{th} round, i.e.,

$$p_u(s) = \Pr(\dim(S_u(s+1)) = i | \dim(S_u(s)) = i) \ .$$

Note that, we have from Lemma 2.1,

$$p_u(s) \leq 1 - (1 - h_u(s))(1 - \tfrac{1}{q}) \ , \tag{8}$$

since the dimension of node u increases with probability at least $(1 - h_u(s))(1 - \frac{1}{q})$ if all the nodes are equally likely to be called with a *pull* mechanism.

We first note that $h_u(t) \leq i/k$ upon using the argument in the proof of Lemma 4.1 (note that this is a worst case assumption when $\dim(S_u(t)) = i$ corresponds to u decoding i messages). We clearly have

$$p_u(t) \leq 1 - (1 - \tfrac{i}{k})(1 - \tfrac{1}{q}) \stackrel{\triangle}{=} \overline{p}_u(t) \ .$$

In other words, $\overline{p}_u(t)$ denotes the maximum probability of not increasing the dimension after the first time the dimension of a node is i.

We now proceed to obtain $h_u(t+1)$ and hence $p_u(t+1)$, i.e. the probability of not increasing the dimension in round $t+1$, given that the node u could not increase its dimension in round t. Let $\mathcal{N}_t$ be the event that the node u fails to increase its dimension at the end of round t.

Claim 4.1. *If $\frac{i}{k} \geq \frac{1}{2}$, then*

$$h_u(t+1) \leq (\tfrac{i}{k})^2(1+\tfrac{1}{q})(1+\epsilon_n) \stackrel{\triangle}{=} \overline{h}_1 \ , \ \ w.p.\ 1-\tfrac{1}{n^3}$$

where $\epsilon_n = 7\sqrt{\ln(n)/n}$, and $\overline{h}_2$ is given by (7). Thus, from (8),

$$p_u(t+1) \leq 1-(1-\overline{h}_1)(1-\tfrac{1}{q}) < p_u(t), \ \ w.p.\ 1-\tfrac{1}{n^3}.$$

Proof. We clearly have,

$$\Pr(v \in E_u(t+1) | v \in E_u(t), \mathcal{N}_t) \leq p_u(t)$$

The above follows from observing that, if $v \in E_u(t)$, then the node v increases its dimension at the end of round t with probability at least $1-p_u(t)$. It follows that,

$$|E_u(t+1)| \mid \mathcal{N}_t \prec_{st} \text{Bin}(|E_u(t)|, p_u(t)) \prec_{st} \text{Bin}(\tfrac{ni}{k}, \overline{p}_u(t)) \ , \tag{9}$$

since $|E_u(t)| \leq ni/k$.

Now, applying a Chernoff bound for Binomial random variable, we have

$$\begin{aligned}
&\Pr(h_u(t+1) \geq \tfrac{i}{k}\overline{p}_u(t)(1+\epsilon)) \\
&= \Pr(|E_u(t+1)| \geq \tfrac{ni}{k}\overline{p}_u(t)(1+\epsilon) | \mathcal{N}_t) \\
&\leq \Pr(\text{Bin}(\frac{ni}{k}, \overline{p}_u(t)) \geq \tfrac{ni}{k}\overline{p}_u(t)(1+\epsilon)) \ \ \text{(from (9))} \\
&\leq \exp(-\tfrac{\epsilon^2 ni\overline{p}_u(t)}{4k})
\end{aligned}$$

The last step is a standard application of Chernoff bound for a Binomial random variable. Note that, $i/k \geq 1/2$ and $\overline{p}_u(t) \geq 1-(1/2)(1-1/q) \geq 1/2$, and hence, by choosing, $\epsilon = \epsilon_n = 7\sqrt{\ln(n)/n}$, we have $\epsilon^2 ni\overline{p}_u(t)/4k \geq 3\ln(n)$ from which it follows that

$$\Pr(h_u(t+1) \leq \tfrac{i}{k}\overline{p}_u(t)(1+\epsilon_n)) \geq 1-\tfrac{1}{n^3}$$

Further, observe the following.

$$\begin{aligned}
& h_u(t+1) \leq \tfrac{i}{k}\overline{p}_u(t)(1+\epsilon_n) \\
\Rightarrow \ \ & h_u(t+1) \leq \tfrac{i}{k}(1-(1-\tfrac{i}{k})(1-\tfrac{1}{q}))(1+\epsilon_n) \\
& \qquad = ((\tfrac{i}{k})^2 + \tfrac{i}{k}(1-\tfrac{i}{k})\tfrac{1}{q})(1+\epsilon_n) \\
& \qquad = (\tfrac{i}{k})^2(1+\tfrac{1}{q})(1+\epsilon_n) = \overline{h}_1 \quad \text{(by (7))}
\end{aligned}$$

Further, as we have argued earlier, we also have that,

$$p_u(t+1) \leq 1 - (1 - h_u(t+1))(1 - \tfrac{1}{q}) < 1 - (1 - \overline{h}_2)(1 - \tfrac{1}{q}) \; .$$

The upper bound is useful so long as $h_u(t+1) < 1$ which is guaranteed if

$$\tfrac{i}{k}(1 + \tfrac{1}{q})(1 + \epsilon_n) < 1$$

which is true as long as

$$\tfrac{i}{k} < (1 - \tfrac{1}{q})(1 - \epsilon_n)$$

which is further implied by

$$\tfrac{i}{k} \leq 1 - 2\epsilon_k \; . \tag{10}$$

The inequality (10) is trivially true in the regime $i \leq k - 14\sqrt{k \ln(k)}$. Claim 4.1 is thus proved. □

The above calculation can be extended to calculate $p_u(t+2)$, i.e., the probability of node u increasing its dimension in round $t+2$ provided it has not done so at the end of round $t+1$. More precisely, we can show that

$$h_u(t+2) \leq \left(\tfrac{i}{k}\right)^4 (1 + \tfrac{1}{q})^3 (1 + \epsilon_n)^3$$

and so on so forth for $t+3, t+4, \ldots$ so long as the worst case upper bound on $h_u(\cdot)$ is larger than $1/2$. Note that, the quantities $\overline{h}_s$ defined in the statement of Lemma are precisely the upper bounds on $h_u(t+s)$.

Thus, we have shown that the node u starts with a success probability (probability that the dimension increases from i) with at least $(1 - \overline{h}_1)(1 - 1/q)$. If node u fails to increase its dimension from after one round of message exchange, with a probability of at least $(1 - 1/n^3)$ the system reaches a state at which the success probability becomes at least $(1 - \overline{h}_2)(1 - 1/q)$, and so on so forth until the success probability becomes $(1/2)(1 - 1/q)$ (which happens when $\overline{h}_s \leq 1/2$). Thus a discrete phase type distribution with transition probability as given in the Lemma provides a stochastic upper bound on the random variable T_i.

□

We now proceed to analyze the regime $i \geq k - 14\sqrt{k \ln(k)}$ when there may not be much benefit provided by a coding based approach. In the regime, $i \geq k - 14\sqrt{k \ln(k)}$, we note than T_i cannot be "worse" than the time taken to disseminate a single message in the whole network using a pull mechanism which is $\ln(n)$ $w.h.p$. We have the following result in this regime.

Lemma 4.3. *For $i \geq k - 14\sqrt{k \ln(k)}$ and n large enough,*

$$T_i \prec_{st} \sum_{j=1}^{s_n} G_j + \sum_{j=1}^{r_n} H_j + \mathit{Geom}(\tfrac{1}{2}(1 - \tfrac{1}{q})) \; ,$$

where G_j's are iid with $G_j \sim$ Geom$(1-\sqrt{0.4})$, and $H_j's$ are iid with $H_j \sim$ Geom(0.4)[2] *, $s_n = \ln(n)$ and $r_n = \frac{\ln(n/2)}{\ln(4/3)}$.*

The proof is almost identical to pull with a single message. We provide the proof in the Appendix for completeness.

4.2 Upper bound on mean and high probability bound

Lemma 4.4. *For $i \leq k/2$,*

$$\mathbb{E}[T_i] \leq \frac{2q}{q-1} .$$

Further

$$\sum_{i=1}^{k/2} T_i \leq \frac{q}{q-1}(k + 4\ln(n) + 2\sqrt{k\ln(n)}), \;\; w.p.\; 1 - O(\tfrac{1}{n^2})$$

The proof follows from Lemma 4.1 by an application of Chernoff bound for Geometric random variables and is relegated to the appendix.

We have a similar resulting the regime $k/2 < i < k - 14\sqrt{k\ln(k)}$.

Lemma 4.5. *For $k/2 < i < k - 14\sqrt{k\ln(k)}$, we have*

$$\mathbb{E}[T_i] \leq \frac{2q}{q-1} + 1 + \log(\ln(2)) - \log\left(\ln(\tfrac{k'}{i})\right) + O(\tfrac{1}{n^2}),$$

where $k' = k - 14\sqrt{k\ln(k)}$. Further,

1. $\displaystyle\sum_{i=k/2}^{k-14\sqrt{k\ln(k)}} \mathbb{E}[T_i] \leq \tfrac{k}{2}(\tfrac{2q}{q-1} + 1 + \log(\ln(2)) + \tfrac{1+\ln(2)}{\ln(2)}) + O(\sqrt{k\ln k})$

2. $\displaystyle\sum_{i=k/2}^{k-14\sqrt{k\ln(k)}} T_i \leq \tfrac{k}{2}(1 + \log(\ln(2)) + \tfrac{1+\ln(2)}{\ln(2)} + \tfrac{2q}{q-1}) + O(\ln(n)) + O(\sqrt{k\ln(n)}), \;\; w.p.\; 1 - O(\tfrac{1}{n^2})$

Again, the proof follows from Lemma 4.2 by standard application of Chernoff bound for Geometric random variables and is relegated to the appendix.

Lemma 4.6. *For $i > k - 14\sqrt{k\ln(k)}$, we have*

1. $\mathbb{E}T_i = O(\ln(n))$

2. $\displaystyle\sum_{i>k-14\sqrt{k\ln(k)}} T_i = O(\sqrt{k\ln(k)}(\ln(n)), \;\; w.p.\; 1 - O(\tfrac{1}{n^2}) .$

[2] Instead of 0.4, any quantity slightly smaller than 0.5 works just as well

The proof the above is relegated to the apendix.

We now prove the following from which Theorem 3.1 follows immediately.

Theorem 4.1. *Using an RLC approach and with $q \geq k$, we have,*

$$\sum_{i=1}^{k} \mathbb{E}[T_i] \leq k(\tfrac{2q}{q-1} + \tfrac{1+\log(\ln(2))+(1+\ln(2))/\ln(2)}{2}) + O(\sqrt{k \ln k}\ln(n)) \leq 3.46k + O(\sqrt{k\ln(k)}\ln(n)) \,,$$

and

$$\overline{T}_{RLC}^{pull} \leq 3.46k + O(\sqrt{k\ln(k)}\ln(n)), \;\; w.p. \; 1 - O(\tfrac{1}{n})$$

Proof. Combine Lemma 4.4, 4.5, and 4.6 to show that

$$\begin{aligned}\sum_i T_i &\leq k(\tfrac{2q}{q-1} + \tfrac{1+\log(\ln(2))+(1+\ln(2))/\ln(2)}{2}) + O(\sqrt{k\ln(k)}\ln(n)) \;\; w.p.\, 1 - O(1/n^2) \\ &\leq 3.46k + O(\sqrt{k\ln(k)}\ln(n)) \;\; w.p.\, 1 - O(1/n^2).\end{aligned}$$

If $\overline{T}^j$ denotes the time required for the j^{th} node to decode all the messages, clearly $\overline{T}^j \sim \sum_i T_i = O(n)$ $w.p.\, 1 - O(1/n^2)$. Further, $\overline{T}_{RLC}^{pull} = \overline{T}^1 \vee \overline{T}^2 \vee \overline{T}^3 \ldots \overline{T}^n$. However, it is easy to argue that

$$\Pr(\overline{T}^1 \vee \overline{T}^2 \vee \overline{T}^3 \ldots \overline{T}^n < x) > \prod_{j=1}^{n} \Pr(\overline{T}^j < x) = \left(\Pr(\sum_{i=1}^{k-1} T_i < x)\right)^n$$

since $\Pr(\overline{T}^1 < x | \overline{T}^2 < x) > \Pr(\overline{T}^1 < x)$ trivially. It follows that, if each of $\overline{T}^j$ satisfies the upper bound $w.p.\, 1 - O(1/n^2)$, $\overline{T}_{RLC}^{pull}$ too satisfies the upper bound $w.p.\, (1 - O(1/n^2))^n = 1 - O(1/n)$. □

5 Random Linear Coding with "Push"

We continue to use the same notation as in Section 4. Again, we denote by T_i the time spent by a node when it has a dimension i.

5.1 Stochastic bounds

We first derive a key lemma which will be used repeatedly in all our proofs in this subsection.

Lemma 5.1. *Suppose $q \geq \max(k, \ln(n))$. Let S_u^- and S_u^+ be the subspaces spanned by the code-vectors with node u at the beginning and the end of a typical round, respectively. At the beginning of the round, let E_u be the set of nodes that cannot help node u, i.e.,*

$$E_u = \{v : S_v^- \subseteq S_u^-\}$$

and let h_u denote the fraction of these nodes, so that $h_u = |E_u|/n$. Then, for all $n \geq n_0$ (for a suitable constant n_0), with a push based mechanism under RLC protocol,

$$\Pr(\dim(S_u^+) = \dim(S_u^-)) \leq \left(\tfrac{2}{5}\right)^{1-h_u}$$

Proof. Note that, if a node $v \in \overline{E}_u$ pushes to node u (which happens with probability $1/n$), then, node u increases its dimension with probability at least $1 - 1/q$ by Lemma 2.1. Thus, $v \in \overline{E}_u$ increases the dimension of u with probability at least $(1/n)(1 - 1/q)$. We thus have,

$$\begin{aligned}
&\Pr(\dim(S_u^+) = \dim(S_u^-)) \\
&= \Pr(\cap_{v \in \overline{E}_u} \{v \text{ does not increase the dimension of node } u\}) \\
&= \big(1 - \Pr(\text{any node } v \in \overline{E}_u \text{ increases the dimension of } u)\big)^{|\overline{E}_u|} \\
&\leq \left(1 - \tfrac{1}{n}(1 - \tfrac{1}{q})\right)^{|\overline{E}_u|} \\
&\leq \left(1 - \tfrac{1}{n} + \tfrac{1}{n\ln(n)}\right)^{n - |E_u|} \quad (\because\ q \geq \ln(n)) \\
&\leq \left(1 - \tfrac{1}{n} + \tfrac{1}{n\ln(n)}\right)^{(n(1-h_u))} .
\end{aligned}$$

It can be shown that the quantity $(1 - 1/n + 1/(n\ln(n))^n$ is decreasing for large n, and further, for n large enough

$$\left(1 - \tfrac{1}{n} + \tfrac{1}{n\ln(n)}\right)^n \leq \tfrac{2}{5}$$

In fact, the quantity $2/5$ can be replaced by any quantity strictly larger than $1/e$. The result thus follows. □

We decompose our analysis into three regimes, $i \leq k/2$, $k/2 < i < k - \sqrt{k}\ln(k)$, and $i \geq k - \sqrt{k}\ln(k)$. For the case $i \leq k/2$, we have a trivial stochastic upper bound on T_i from Lemma 2.1 and Lemma 5.1.

Lemma 5.2. *For $i \leq k/2$, and n large enough (i.e., $\forall n \geq n_0$ for some n_0),*

$$T_i \prec_{st} Y_i \ ,$$

where,

$$Y_i \sim \textit{Geom}\left(1 - \sqrt{\tfrac{2}{5}}\right) \ ,$$

Proof. Consider a time instant t and let $S_u(t)$ be the subspace spanned by the code-vectors received by node u at the beginning of round t. Let $\dim(S_u(t)) = i$ and let $E_u(t)$ be the nodes that cannot help u, i.e.,

$$E_u(t) = \{v : S_v(t) \subseteq S_u(t)\}$$

and let $h_u(t) = |E_u(t)|/n$. Using an argument, similar to the one in the proof of Lemma 4.1, we have

$$h_u(t) \leq \frac{i}{k}$$

For n large enough, it immediately follows using Lemma 5.1 that

$$\Pr(\dim(S_u(t+1)) = \dim(S_u(t))) \leq \left(\tfrac{2}{5}\right)^{1-h_u} \leq \left(\tfrac{2}{5}\right)^{((k-i)/k)} \leq \sqrt{\tfrac{2}{5}}$$

□

In the regime $k/2 < i < k - \sqrt{k}\ln(k)$, we have the following.

Lemma 5.3. *For $k/2 < i \leq k - \sqrt{k}\ln(k)$,*

$$T_i \prec_{st} Y_i \ ,$$

where, Y_i is a discrete phase type distribution on the finite state space $\{1, 2, 3, \ldots \overline{s}_i\}$ with transition probability structure given by

$$p_{s,s+1} = (1 - \tfrac{1}{n^3})(1 - p_{s,\mathcal{A}}), \quad s \leq \overline{s}_i \tag{11}$$

$$p_{s,\mathcal{A}} = 1 - \left(\tfrac{2}{5}\right)^{1-\overline{h}_s}, \ s \leq \overline{s}_i \tag{12}$$

$$p_{\overline{s}_i,\mathcal{A}} = 1 - p_{\overline{s}_i,\overline{s}_i} = 1 - \sqrt{\tfrac{2}{5}}, \tag{13}$$

where

$$\overline{h}_s = 1 - (1+\gamma)^s \left(\tfrac{k-i}{k}\right), \quad (\gamma = 0.2^3)$$

and

$$\overline{s}_i = \min\{s : \overline{h}_s \leq \tfrac{1}{2}\} \ .$$

The Markov Chain corresponding to the discrete phase type distribution of Y_i is shown in Figure 5.

Proof. The proof starts along the lines of the proof of Lemma 4.2 but there are some key differences because we are dealing with a *push* mechanism here. As before, let t be the first time at which the node u has dimension i. Let $T_i + t$ be the first time when the dimension of of the node u increases from i.

We continue to use the notation as before. To remind the reader, let $S_v(s)$ be the subspace spanned by the coded messages with node v at the beginning of round s. Let $E_u(s)$ be the nodes which cannot help u if they call u at the beginning of round s, i.e.,

$$E_u(s) = \{v : S_v(s) \subseteq S_u(s)\}$$

[3] In fact any quantity slightly smaller than $\frac{\ln(5/2)}{4}$ serves our purpose

We also denote the fraction of such nodes by $h_u(s)$ so that $h_u(s) = |E_u(s)|/n$. We denote by $p_u(s)$ the probability that the dimension of node u remains i after one round of message exchange in the s^{th} round, i.e.,

$$p_u(s) = \Pr(\dim(S_u(s+1)) = i | \dim(S_u(s)) = i) \ .$$

We first note that $h_u(t) \leq i/k$ upon using the argument in the proof of Lemma 4.1 (note that this is a worst case assumption when $\dim(S_u(t)) = i$ corresponds to u decoding i messages). We have from Lemma 5.1 that,

$$p_u(t) \leq \left(\tfrac{2}{5}\right)^{1-h_u(t)} \leq \left(\tfrac{2}{5}\right)^{1-(i/k)} \triangleq \overline{p}_u(t) \ .$$

We now proceed to obtain $h_u(t+1)$ and hence $p_u(t+1)$, i.e., given that the node u could not increase its dimension in round t, the probability of not increasing the dimension in round $t+1$. Let $\mathcal{N}_t$ be the event that the node u does not increase its dimension at the end of round t. We need to introduce some notation here. Let $r = |E_u(t)|$ and $y = n - r$. Let us index the nodes in $E_u(t)$ by $l_1, l_2 \ldots l_r$ where $r = |E_u(t)|$. We introduce the following $0-1$ random variables.

$$Z_{l_i} = \begin{cases} 1, & \text{if } l_i \in E_u(t+1) \ | l_i \in E_u(t), \\ 0, & \text{else, i.e., if } l_i \in \overline{E}_u(t+1) \ | l_i \in E_u(t). \end{cases}$$

Note that, $|E_u(t+1)| = \sum_{j=1}^{|E_u(t)|} Z_{l_j}$. However, the random variables Z_{l_j}'s are not independent, unlike the "pull" case. However, Z_{l_j}'s are negatively correlated and so we can still apply the Chernoff bound for appropriate Binomial random variables. More precisely, we may prove the following.

Claim 5.1. *For all $n \geq n_0$ (for a suitable n_0),*

$$\mathbb{E}\left[\exp(\theta \sum_{j=1}^{r} Z_{l_j})\right] \leq \mathbb{E}\left[\exp(\theta X)\right] \ , \ \ \forall \, \theta > 0$$

where,

$$X \sim \textit{Bin}\left(r, \overline{p}_u(t)\right)$$

It thus follows that a Chernoff-bound on the upper-tail of $\sum_{j=1}^{r} Z_{l_j}$ can be bounded from above by a Chernoff-bound on the upper tail of X.

Proof. We want to show that the random variables Z_{l_j}'s are negatively correlated. To this end, we first compute $\mathbb{E}[Z_{l_m} Z_{l_j}]$. First, we note that

$$\mathbb{E}[Z_{l_j}] = \left(1 - \tfrac{1}{n} + \tfrac{1}{nq}\right)^y$$

upon using the argument in the proof of Lemma 5.1. Next note that,

$$\begin{aligned}
\mathbb{E}[Z_{l_m} Z_{l_j}] &= \Pr(Z_{l_m} = 1, Z_{l_j} = 1) \\
&= \sum_{r_1+r_2 \le y} \Pr(l_m \text{ is pushed by } r_1 \text{ nodes in } \overline{E}_u(t),\ l_j \text{ is pushed by } r_2 \text{ nodes in } \overline{E}_u(t)) \\
&\times \Pr(Z_{l_m} = 1, Z_{l_j} = 1 \mid l_m \text{ is pushed by } r_1 \text{ nodes in } \overline{E}_u(t),\ l_j \text{ is pushed by } r_2 \text{ nodes in } \overline{E}_u(t)) \\
&\le \sum_{r_1+r_2 \le y} \Pr(l_m \text{ is pushed by } r_1 \text{ nodes in} \overline{E}_u(t),\ l_j \text{ is pushed by } r_2 \text{ nodes in} \overline{E}_u(t)) \frac{1}{q^{r_1+r_2}}
\end{aligned}$$

Observe that,

$$\begin{aligned}
&\Pr(l_m \text{ is pushed by } r_1 \text{ nodes in } \overline{E}_u(t),\ l_j \text{ is pushed by } r_2 \text{ nodes in } \overline{E}_u(t)) \\
&= \frac{y!}{r_1! r_2! (y - r_1 - r_2)!} \frac{(n-2)^{y-(r_1+r_2)}}{n^y}
\end{aligned}$$

which follows from simple combinatorial considerations. It follows that,

$$\begin{aligned}
\mathbb{E}[Z_{l_m} Z_{l_j}] &\le \sum_{r_1+r_2 \le y} \frac{y!}{r_1! r_2! (y - r_1 - r_2)!} \left(1 - \frac{2}{n}\right)^{y-(r_1+r_2)} \frac{1}{(nq)^{r_1+r_2}} \\
&= \left(1 - \tfrac{2}{n} + \tfrac{2}{nq}\right)^y \quad \text{(using multinomial expansion)} \\
&\le \left(1 - \tfrac{1}{n} + \tfrac{1}{nq}\right)^{2y} .
\end{aligned}$$

We thus have,

$$\begin{aligned}
\mathbb{E}[\sum_{j=1}^{r} Z_{l_j}]^2 &\le r\mathbb{E}[Z_{l_1}^2] + r(r-1)\mathbb{E}[Z_{l_1} Z_{l_2}] \\
&\le r\left(1 - \tfrac{1}{n} + \tfrac{1}{nq}\right)^y + r(r-1)\left(1 - \tfrac{1}{n} + \tfrac{1}{nq}\right)^{2y} \\
&= \mathbb{E}[\sum_{j=1}^{r} X_{l_j}]^2 ,
\end{aligned}$$

where X_{l_j}'s are iid $0-1$ Bernoulli random variables with mean $(1 - 1/n + 1/(nq))^y$. A very similar calculation yields,

$$\mathbb{E}[Z_{l_1} Z_{l_2} Z_{l_3} \dots Z_{l_m}] \le \left(1 - \tfrac{1}{n} + \tfrac{1}{nq}\right)^{my}, \ \ \forall\, m \ge 1 \tag{14}$$

Further, since Z_{l_j}'s are $0-1$ random variable, it follows from (14) that, for all positive integers c,

$$\mathbb{E}[\sum_{j=1}^{r} Z_{l_j}]^c \le \mathbb{E}[\sum_{j=1}^{r} X_{l_j}]^c ,$$

where X_{l_j} are iid $0-1$ Bernoulli random variables with mean $(1-1/n+1/(nq))^y$. Thus $\mathbb{E}[\exp(\theta \sum_{j=1}^{r} Z_{l_j})] \leq \mathbb{E}[\exp(\theta \sum_{j=1}^{r} X_{l_j})]$ for $\theta \geq 0$. We further have

$$\begin{aligned}
&\left(1 - \tfrac{1}{n} + \tfrac{1}{nq}\right)^y \\
&\leq \left(1 - \tfrac{1}{n} + \tfrac{1}{n\ln(n)}\right)^{(n(k-i)/k)} \\
&\leq \left(\tfrac{2}{5}\right)^{(k-i)/k}, \text{ for } n \text{ large enough}
\end{aligned}$$

and thus $\sum_{j=1}^{r} X_{l_j} \prec_{st} \text{Bin}(r, (\frac{2}{5})^{(k-i)/k})$. Claim 5.1 thus follows. □

We now have the following bound on $h_u(t+1)$.

Claim 5.2. *For n large enough and $\frac{i}{k} \geq \frac{1}{2}$,*

$$h_u(t+1) \leq 1 - (1+\gamma)^{\frac{k-i}{k}}, \;\; w.p.\; 1 - \tfrac{1}{n^3}$$

where $\gamma = 0.2$. It follows from Lemma 5.1 that,

$$p_u(t+1) \leq \left(\tfrac{2}{5}\right)^{(1+\gamma)\frac{k-i}{k}}$$

Proof. We have that

$$|E_u(t+1)| \mid \mathcal{N}_t \sim \sum_{j=1}^{r} Z_{l_j}$$

Now, applying a Chernoff bound for Binomial random variable, we have,

$$\begin{aligned}
&\Pr(|E_u(t+1)| \geq \tfrac{ni}{k}\overline{p}_u(t)(1+\epsilon)|\mathcal{N}_t) \qquad (15)\\
&\leq \inf_{\theta>0} \frac{\mathbb{E}(\exp(\theta \sum_{j=1}^{r} Z_{l_j})}{\exp(\theta \frac{ni}{k}\overline{p}_u(t)(1+\epsilon))} \\
&\leq \inf_{\theta>0} \frac{\mathbb{E}(\exp(\theta \sum_{j=1}^{ni/k} X_{l_j})}{\exp(\theta \frac{ni}{k}\overline{p}_u(t)(1+\epsilon))} \quad \text{(appealing to Claim 5.1, } \sum_j X_{l_j} \sim \text{Bin}(\tfrac{ni}{k}, \overline{p}_u(t))) \\
&\leq \exp(-\tfrac{\epsilon^2 ni \overline{p}_u(t)}{4k})
\end{aligned}$$

where the last step follows from the fact that the infimizing θ in the second last step precisely corresponds to the Chernoff bound for $\Pr(\text{Bin}(ni/k, \overline{p}_u(t)) \geq \frac{ni}{k}\overline{p}_u(t)(1+\epsilon))$. Note that, $i/k \geq 1/2$ and $\overline{p}_u(t) = (2/5)^{(k-i)/k} \geq \sqrt{2/5}$ (since $i > k/2$), and hence, by choosing, $\epsilon = \epsilon_n = 7\sqrt{\ln(n)/n}$, we have $\epsilon^2 n \overline{p}_u(t)/2 \geq 3\ln(n)$ from which it follows that

$$\Pr(h_u(t+1) \leq \tfrac{i}{k}\overline{p}_u(t)(1+\epsilon_n)) \geq 1 - \tfrac{1}{n^3}$$

Further observe the following.

$$
\begin{aligned}
& h_u(t+1) \leq \tfrac{i}{k}\overline{p}_u(t)(1+\epsilon_n) \\
\Rightarrow \quad & h_u(t+1) \leq (1-\tfrac{k-i}{k})\left(\tfrac{2}{5}\right)^{(k-i)/k}(1+\epsilon_n) \\
& \leq \frac{1-\frac{k-i}{k}}{1+\beta\frac{k-i}{k}}(1+\epsilon_n) \quad \left(\beta = \ln(\tfrac{5}{2}),\ \because\ \left(\tfrac{2}{5}\right)^{(k-i)/k} < \frac{1}{1+\ln(\frac{5}{2})\frac{k-i}{k}}\right) \\
& \leq (1-\tfrac{k-i}{k})(1-\tfrac{\beta}{2}\tfrac{k-i}{k})(1+\epsilon_n) \quad (\because\ \tfrac{1}{1+x} \leq 1-\tfrac{x}{2} \text{ for } 0 \leq x \leq 1) \\
& \leq (1-\tfrac{k-i}{k}(1+\tfrac{\beta}{2})+\tfrac{\beta}{2}(\tfrac{k-i}{k})^2)(1+\epsilon_n) \\
& \leq (1-\tfrac{k-i}{k}(1+\tfrac{\beta}{2}-\tfrac{\beta}{4}))(1+\epsilon_n) \quad (\because\ \tfrac{k-i}{k} \leq \tfrac{1}{2}) \\
& \leq 1-(1+\tfrac{\beta}{4})\tfrac{k-i}{k}+\tfrac{k-i}{k}\tfrac{k}{k-i}\epsilon_n \\
& \leq 1-(1+\tfrac{\beta}{4})\tfrac{k-i}{k}+\tfrac{k-i}{k}\tfrac{k}{\sqrt{k}\ln(k)}\epsilon_n \quad (\because\ k-i \geq \sqrt{k}\ln(k)) \\
& \leq 1-(1+\tfrac{\beta}{4})\tfrac{k-i}{k}+\tfrac{k-i}{k}\tfrac{7}{\sqrt{\ln(n)}} \\
& \leq 1-(1+\tfrac{\beta}{4}-\delta)\left(\tfrac{k-i}{k}\right),
\end{aligned}
$$

where δ can be chosen as arbitrarily small for large enough n. More precisely, since $\beta/4 \approx 0.24$, we can choose n large enough so that $\beta/4 - \delta = \gamma = 0.2$, say. We thus have,

$$
\begin{aligned}
& h_u(t+1) \leq \tfrac{i}{k}\overline{p}_u(t)(1+\epsilon_n) \\
\Rightarrow \quad & h_u(t+1) \leq 1-(1+\gamma)\tfrac{k-i}{k} \quad (\gamma = 0.2)
\end{aligned} \tag{16}
$$

Claim 5.2 thus follows. □

In effect, we have shown that, if node u fails to increase its dimension from i after one more round of message exchange, with a probability of at least $(1-1/n^3)$ the system reaches a state in which $1-h_u(t+1) \geq (1+\gamma)(k-i)/k$ and the corresponding success probability in round $t+1$ immediately follows from Lemma 5.1. If the dimension of the node u does not increase at the end of round $t+1$, a similar calculation yields $1 - h_u(t+2) \geq (1+\gamma)^2(k-i)/k$ with probability at least $1 - 1/n^3$, and so on so forth until the upper bound on $h_u(\cdot)$ is less than $1/2$. Thus a discrete phase type distribution with the given transition probabilities provides a stochastic bound (in a stochastic order sense) on the random variable T_i. □

In the regime, $i \geq k - \sqrt{k}\ln(k)$, we note than T_i cannot be "worse" than the time taken to disseminate a single message in the whole network using a pull mechanism which is $\ln(n)$ $w.h.p$. We have the following result in this regime.

Lemma 5.4. *For $i \geq k - \sqrt{k}\ln(k)$,*

$$T_i \prec_{st} \sum_{j=1}^{s_n} H_j + \textit{Geom}(1 - \sqrt{\tfrac{2}{5}}) \ ,$$

where $G_j's$ are iid with distribution Geom(0.5), $H_j's$ are iid with distribution Geometric(θ) for some $\theta \in (0,1)$ and for large enough n. Further $r_k = \log_{(1+\gamma)}(n)$ ($\gamma = 0.2$).

The proof is not much different from that of a single message with a *push* mechanism. We provide a sketch of the proof in the Appendix, which is slightly different from the proof of Lemma 4.3.

5.2 Upper bound on mean and high probability bound

Lemma 5.5. *For $i \leq k/2$,*

$$\sum_{i=1}^{k/2} \mathbb{E}[T_i] \leq \frac{k/2}{1 - \sqrt{2/5}} \leq 1.36k \ .$$

Further

$$\sum_{i=1}^{k/2} T_i = \frac{q}{q-1}(k + 4\ln(n) + 2\sqrt{k\ln(n)}), \ \ w.p.\ 1 - O(\tfrac{1}{n^2})$$

Proof. Similar to the proof of Lemma 4.4 □

We have the following in the regime $k/2 < i < k - \sqrt{k}\ln(k)$.

Lemma 5.6. *For $k/2 < i < k - \sqrt{k}\ln(k)$, we have*

$$\mathbb{E}[T_i] \leq \frac{1}{1 - \sqrt{\frac{2}{5}}} + 1 + \frac{\ln(k) - \ln(2(k-i))}{\ln(1+\gamma)} + O(\tfrac{1}{n^2})$$

Further,

1. $\displaystyle\sum_{i=k/2}^{k-\sqrt{k}\ln(k)} \mathbb{E}T_i \leq \tfrac{k}{2}(\tfrac{1}{1-\sqrt{2/5}} + 1 + \tfrac{1}{\ln(1+\gamma)}) \leq 4.6k$

2. $\displaystyle\sum_{i=k/2}^{k-\sqrt{k}\ln(k)} T_i = 4.6k + O(\ln(n)) + O(\sqrt{k\ln(n)}), \ \ w.p.\ 1 - O(\tfrac{1}{n^2})$

Proof. Please see the proof of Lemma 4.5. □

Lemma 5.7. *For $i > k - \sqrt{k}\ln(k)$, we have*

1. $\mathbb{E}T_i = O(\ln(n))$

2. $\sum_{i>k-\sqrt{k}\ln(k)} T_i = O(\sqrt{k}\ln(k)\ln(n)),\ \ w.p.\ 1 - O(\frac{1}{n^2})$.

Proof. Please see the proof of Lemma 4.6 □

We now have all the ingredients to prove Theorem 3.2, which we restate below for convenience.

Theorem 5.1. *Using an RLC approach, we have,*

$$\mathbb{E}T_{RLC}^{push} \leq 5.96k + O(\sqrt{k}\ln(k)\ln(n)) ,$$

and

$$\overline{T}_{RLC}^{push} = 5.96k + O(\sqrt{k}\ln(k)\ln(n)),\ \ w.p.\ \ 1 - O(\tfrac{1}{n})$$

Proof. Combine Lemma 5.5, 5.6, and 5.7 with minor additional arguments as in the proof of Theorem 4.1. □

6 Random Message Selection with "Pull"

In this case, we divide the time required for all the nodes to receive all the messages into $k-1$ phases, where the i^{th} phase corresponds to the minimum number of messages received by the nodes in the network being i. Let T_i be the random variable denoting the time spent in phase i. Denote by T_{RMS} the total time taken by by all the nodes to receive all the messages. Note that, phase in this case refers to state of all the nodes, whereas, phase in the RLC approach refers to a state of any particular node.

Lemma 6.1. *For $i > k/2$,*

$$T_i \succ_{st} \textit{Geom}\left(\tfrac{k-i}{i}\right)$$

Proof. Let S_v^- denote the set of all the messages with v at the beginning of a round. Let $i = \min_{v\in[n]} |S_v^-|$ and WLOG $\arg\min_{v\in[n]} |S_v^-| = u$. Let

$$p_s = \Pr(|S_u^+| > |S_u^-|) ,$$

where S_u^+ is the set of messages with the node u after one more round of message exchange. Note that,

$$\begin{aligned} p_s &= \frac{1}{n}\sum_{v\in[n]} \frac{|S_v^- \setminus S_u^-|}{|S_v^-|} \\ &= 1 - \frac{1}{n}\sum_{v\in[n]} \frac{|S_v^- \cap S_u^-|}{|S_v^-|} \end{aligned}$$

where the first step follows from the fact that each node transmits any of the messages it has with equal probability in an RMS approach. Further since,

$$|S_v^- \cap S_u^-| \geq |S_v^-| + |S_u^-| - k$$

we have

$$\begin{aligned}\frac{|S_v^- \cap S_u^-|}{|S_v^-|} &\geq \frac{|S_v^-| + |S_u^-| - k}{|S_v^-|} \\ &= 1 - \frac{k-i}{|S_v^-|} \\ &\geq 1 - \frac{k-i}{i}\end{aligned}$$

where the last step follows using $|S_v^-| \geq i$ in the i^{th} phase. Clearly,

$$p_s \leq \min(1, \tfrac{k-i}{i}) \; .$$

Since, $p_1 \leq p_2$ implies Geom$(p_2) \prec_{st}$ Geom(p_1), the result follows. □

We are now in a position to prove Theorem 3.3 which we restate below.

Theorem 6.1. *Suppose $k = \alpha n$. We then have*

1. $\sum_{i=k/2}^{k-1} \mathbb{E}T_i = \Omega(n \ln n)$
2. $\lim_{k\to\infty} \Pr\left(\sum_{i=k/2}^{k-1} T_i = \Omega(k \ln(k))\right) = 1$

Proof. Note that,

$$\sum_{i=k/2}^{k-1} \frac{i}{k-i} = -\frac{k}{2} + k \sum_{i=k/2}^{k-1} \frac{1}{k-i} \approx \frac{k \ln(k)}{2} \; .$$

which is $\Omega(n \ln n)$ (since $n = \alpha k$). For the second part, consider the random variable

$$S_k = \frac{\sum_{i=k/2}^{k-1} (Y_i - \mathbb{E}Y_i)}{k \ln(k)} \; ,$$

where $Y_i \sim Geom((k-i)/i)$ and Y_i's are independent. As $T_i \succ_{st} Y_i$ from Lemma 6.1, it follows that

$$\text{Var}(Y_i) = (\tfrac{i}{k-i})^2 - \tfrac{i}{k-i}$$

from which it can be shown through some algebraic manipulations that

$$\sum_{i=k/2}^{k-1} \text{Var}(Y_i) \leq \xi k^2$$

for a suitable constant $\xi > 0$. We thus have,

$$\text{Var}(S_k) \leq \frac{\xi}{(\ln(k))^2} \to 0$$

and hence $S_k \to 0$ *in probability* as $k \to \infty$. We thus have for any given $\delta > 0$,

$$\Pr(\sum_i Y_i \geq \sum_i \mathbb{E}[Y_i] - \delta k \ln(k)) \to 1$$

and the result follows since $\sum_i T_i \succ_{st} \sum_i Y_i$. □

7 Random Message Selection with "Push"

In this case, we divide the time required for all the nodes to receive all the messages into $k-1$ phases, where the i^{th} phase corresponds to the minimum number of messages received by the nodes in the network being i. Let T_i be the random variable denoting the time spent in phase i. Denote by T_{RMS} the total time taken by all the nodes to receive all the messages. Note that, phase in this case refers to state of all the nodes, whereas, phase in the RLC approach refers to a state of any particular node.

Lemma 7.1. *For $i > k/2$,*

$$T_i \succ_{st} \textit{Geom}\left(\tfrac{k-i}{i}\right)$$

Proof. The proof follows along the similar lines as in the proof of Lemma 6.1. We simply point the minor differences in the argument due to the "push" mechanism.

As before, let S_v^- denote the set of all the messages with v at the beginning of a round. Let $i = \min_{v\in[n]} |S_v^-|$ and WLOG $\arg\min_{v\in[n]} |S_v^-| = u$. Let

$$p_s = \Pr(|S_u^+| > |S_u^-|)\ ,$$

where S_u^+ is the set of messages with the node u after one more round of message exchange. Further, let p_v denote the probability that node u gets a new message from node v. Clearly,

$$p_v = \frac{1}{n}\frac{|S_v^- \setminus S_u^-|}{|S_v^-|}\ ,$$

since node v calls node u with probability $1/n$. Using an argument similar to the proof of Lemma 6.1, we have

$$p_v = \frac{1}{n}\frac{|S_v^- \setminus S_u^-|}{|S_v^-|} \leq \frac{1}{n}\frac{k-i}{i}$$

Further, note that,

$$\begin{aligned} p_s &= 1 - \prod_{v \in [n]} (1 - p_v) \\ &\leq 1 - \prod_{v \in [n]} (1 - \tfrac{k-i}{in}) \\ &= 1 - (1 - \tfrac{k-i}{in})^n \\ &\leq \tfrac{k-i}{i} \quad (\because (1-x)^n \geq 1 - nx, \text{ for } 0 < x < 1) \end{aligned}$$

The result thus follows from the fact that, $p_1 \leq p_2$ implies $\text{Geom}(p_2) \prec_{st} \text{Geom}(p_1)$. □

We have the following result which can be proved in exactly the same manner as Theorem 6.1.

Theorem 7.1. *Suppose $k = \alpha n$. We then have*

1. $\sum_{i=k/2}^{k-1} \mathbb{E}T_i = \Omega(n \ln n)$
2. $\lim_{k \to \infty} \Pr\left(\sum_{i=k/2}^{k-1} T_i = \Omega(k \ln(k))\right) = 1$

8 Concluding Remarks

We considered the problem of disseminating multiple messages simultaneously in a large network using *gossip-based* dissemination mechanisms. We have presented a protocol based on *random linear coding* that spreads the messages in optimal time in an order sense. The RLC protocol is quite general and does not depend on the underlying communication model. However, we have demonstrated the benefits of the protocol over a *gossip-based* communication model and in a worst case demand scenario when all the nodes want everything. There are a few avenues one might pursue for future research. One avenue for research is to derive results similar to the ones in this paper when the underlying communication graph is not complete. A good starting point might be understand the gains due to simultaneous dissemination in sparse graphs. Another path to tread might be to design protocols when there are some malicious nodes in the network. The application of *random linear coding* for security has been demonstrated in [19] in a different setting.

Acknowledgments

It is a pleasure to thank Dr. Tracey Ho and Prof. David Karger for helpful discussions.

References

[1] S. Accendanski, S. Deb, M. Médard, and R. Koetter. How good is random linear coding based distributed networked storage? In *Proceedings of First Workshop on Network Coding, WiOpt 2005*, April 2005.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46:1024–1016, 2000.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D.Shah. Gossip and mixing times of random walks on random graphs. Preprint available at http://www.stanford.edu/b̃oyd/gossip_gnr.html.

[4] A. Demers et. al. Epidemic algorithms for replicated database maintenance. In *Proc. ACM Symposium on Principles of Distributed Computing*, 1987.

[5] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. J. Wiley & Sons, New York, 1964.

[6] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. J. Wiley & Sons, New York, 1964.

[7] T. Ho, M. Médard, M. Effros, and D. Karger. The benefits of coding over routing in a randomized setting. In *Proc. IEEE Symposium on Information Theory*, 2003.

[8] T. Ho, M. Médard, M. Effros, and D. Karger. On randomized network coding. In *Proc. 41st Allerton Annual Conference on Communication, Control and Computing*, October 2003.

[9] Network Coding Homepage. http://www.comm.csl.uiuc.edu/koetter/nwc/.

[10] S. Jaggi, P.A. Chou, and K. Jain. Low complexity algebraic network codes. In *Proceedings of the IEEE International Symposium on Information Theory*, 2003.

[11] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. Foundations of Computer Science*, 2000.

[12] D. Kempe and J. Kleinberg. Protocols and impossibility results for gossip-based communication mechanisms. In *Proc. 43rd IEEE Symposium on Foundations of Computer Science*, 2002.

[13] David Kempe, Jon M. Kleinberg, and Alan J. Demers. Spatial gossip and resource location protocols. In *Proc. ACM Symposium on Theory of Computing*, 2001.

[14] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, October 2003.

[15] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, Februray 2003.

[16] Y. Minski. Spreading rumors cheaply, quickly, and reliably, 2002. Ph.D. Thesis, Cornell University.

[17] S. M. Ross. *Stochastic processes*. J. Wiley & Sons, New York, 1983.

[18] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In *15th ACM Symposium on Parallel Algorithms and Architectures*, pages 286–294, 2003.

[19] R. Koetter M. Médard T. Ho, B. Leong and M. Effros. Byzantine modification detection in multicast networks using randomized network coding. In *IEEE International Symposium on Information Theory*, June 2004.

A Proofs of Lemme

A.1 Proof of Lemma 4.3

Lemma A.1. *For $i \geq k - 14\sqrt{k\ln(k)}$ and n large enough,*

$$T_i \prec_{st} \sum_{j=1}^{s_n} G_j + \sum_{j=1}^{r_n} H_j + \mathit{Geom}(\tfrac{1}{2}(1-\tfrac{1}{q}))\ ,$$

where G_j's are iid with $G_j \sim \mathit{Geom}(1-\sqrt{0.4})$, and $H_j's$ are iid with $H_j \sim \mathit{Geom}(0.4)$[4] *, $s_n = \ln(n)$ and $r_n = \frac{\ln(n/2)}{\ln(4/3)}$.*

Proof. The proof is not much different from that of a single message with a *pull* mechanism. We provide an outline of the proof below.

Suppose at a particular time t, node u has dimension i in the regime considered in this lemma. We call a node helpful if the subspace spanned by the code-vectors at the node does not lie in that at u, and unhelpful otherwise. Note that, since the node u does not have full rank, there is at least one node that can potentially help node u if called by node u. Starting with a worst case scenario as in the proof of Lemma 4.2, assume that all the other $n-2$ nodes are unhelpful to u. The system then proceeds in rounds. We find the time required until there are at least $n/2$ nodes that can potentially help u. Since u may increase its dimension

[4] Instead of 0.4, any quantity slightly smaller than 0.5 works just as well

before that happens, this time provides a bound on the time u spends at dimension i. We thus consider the following three phases in the evolution.

1. The first phase is until there are at least $\ln(n)$ nodes that can help u, provided the dimension of u remains i throughout this phase. Note that, starting with at least one node that can help u, if p_f is the probability that the number of nodes that cannot help u does not increase after one more round of message exchange, then

$$\begin{aligned}
p_f &= \Pr(\text{none of the unhelpful nodes become helpful}) \\
&\leq \left(1 - \tfrac{1}{n}(1 - \tfrac{1}{q})\right)^{n-1} \\
&\quad (\because \text{ any unhelpful node can potentially increase its dimension } \textit{w.p.} \text{ at least } \tfrac{1}{n}(1 - \tfrac{1}{q})) \\
&\leq \exp\left(-\tfrac{1}{n}(1 - \tfrac{1}{q})(n-1)\right) \\
&\leq (0.4)^{1-1/q} \ \ (\text{for large enough } n) \\
&\leq \sqrt{0.4}.
\end{aligned}$$

   Thus, the probability, p_s, that there is at least one more helpful node after one more round of message exchange is $1 - \sqrt{0.4}$. It follows that, the time spent in this phase is stochastically at most $\sum_{j=1}^{\ln(n)} G_j$, where G_j's are i.i.d. and $G_j \sim \text{Geom}(1 - \sqrt{0.4})$.

2. The second phase starts when there are at least $\ln(n)$ nodes that can potentially help u (if u still has dimension i) and ends when there are at least $n/2$ nodes that can help u. Say, after a few rounds in this phase, there are r nodes that can potentially help node u if called. Note that, $\ln(n) \leq r \leq n/2$ in this phase. We next show that,

$$\Pr(\text{Bin}(n - r, \tfrac{r}{n}(1 - \tfrac{1}{q})) \geq \tfrac{r}{3}) > \tfrac{1}{2} - 0.1 \,, \tag{17}$$

   for large enough n so long as $r \leq n/2$. To see the above, note that,

$$\frac{r}{3} \leq r(1 - \tfrac{r}{n})(1 - \tfrac{1}{k}) \,, \tag{18}$$

   for $k \geq 3$, from which it follows that

$$\begin{aligned}
&\Pr(\text{Bin}(n - r, \tfrac{r}{n}(1 - \tfrac{1}{q})) \geq \tfrac{r}{3}) \\
&\geq \Pr(\text{Bin}(n - r, \tfrac{r}{n}(1 - \tfrac{1}{k})) \geq \tfrac{r}{3}) \quad (\because \ q \geq k) \\
&\geq \Pr(\text{Bin}(n - r, \tfrac{r}{n}(1 - \tfrac{1}{k})) \geq r(1 - \tfrac{r}{n})(1 - \tfrac{1}{k})) \quad (\text{using } (18)) \\
&= \frac{1}{2} - 0.1
\end{aligned}$$

for sufficiently large n. The last step is a simple application of the classic Berry-Esseen theorem [6] which when applied to a sum of independent Bernoulli trials with success probability p boils down to

$$\left|\Pr\left(\frac{\text{Bin}(n,p)-np}{\sqrt{np(1-p)}} \geq x\right) - \Phi(x)\right| \leq \frac{C(p(1-p)^3+(1-p)p^3)}{\sqrt{n}(\sqrt{p(1-p)})^3} \leq \frac{2C}{\sqrt{np(1-p)}}\,,$$

where $\Phi(x)$ is the cdf of a standard normal distribution and C is a universal constant independent of x, n, and p. In our case, we apply the above to $\text{Bin}(n-r, \frac{r}{n}(1-\frac{1}{n}))$ and $x = 0$ which readily implies

$$\begin{aligned}
&|\Pr\left(\text{Bin}(n-r, \tfrac{r}{n}(1-\tfrac{1}{k})) \geq r(1-\tfrac{r}{n})(1-\tfrac{1}{k})\right) - \tfrac{1}{2}| \\
&\leq \frac{2C}{\sqrt{(n-r)\frac{r}{n}(1-\frac{1}{k})(1-\frac{r}{n}+\frac{r}{nk})}} \\
&\leq \frac{2C}{(1-\frac{r}{n})\sqrt{r(1-\frac{1}{k})}} \\
&\leq \frac{4C}{\sqrt{\ln(n)(1-\frac{1}{k})}} \quad (\because\ \ln(n) \leq r \leq n/2) \\
&\leq 0.1 \quad (\text{for large enough } n)
\end{aligned}$$

We have shown (17), and hence the number of nodes that can potentially help node u gets multiplied by $4/3$ with a probability at least 0.4 in each round until $r \leq n/2$. This is assuming that node u has not increased its dimension. Thus, the sum of at most $\ln(n/2)/\ln(4/3)$ geometric random variables with parameter 0.4 takes the system to a state at which there are at least $n/2$ nodes that can potentially help node u when called. The time spent in this phase is thus $\sum_{j=1}^{\ln(n/2)/\ln(4/3)} H_i$ where H_i's are iid and $H_i \sim \text{Geom}(0.4)$.

3. The third phase starts when there are at least $n/2$ nodes that can potentially help u if u has dimension still i. But, then the probability that u increases the dimension is at least $(1/2)(1-1/q)$.

Hence the result by combining the three phases. □

A.2 Proof of Lemma 4.4

Lemma A.2. *For $i \leq k/2$,*

$$\mathbb{E}[T_i] \leq \frac{2q}{q-1}\,.$$

Further

$$\sum_{i=1}^{k/2} T_i \leq \frac{q}{q-1}(k + 4\ln(n) + 2\sqrt{k\ln(n)}),\ \ w.p.\ 1 - O(\tfrac{1}{n^2})$$

Proof. First part of the lemma regarding $\mathbb{E}T_i$ follows immediately from Lemma 4.1. For convenience, define $\mu = 2q/(q-1)$. For the second part, we first observe that $T_i \prec_{st} Y_i$ for $1 \le i \le k/2$ implies that [17]

$$\sum_{i=1}^{k/2} T_i \prec_{st} \sum_{i=1}^{k/2} Y_i$$

from which it follows

$$\begin{aligned}
&\Pr(\sum_{i=1}^{k/2} T_i \ge \tfrac{k\mu}{2}(1+\delta)) \\
&\le \Pr(\sum_{i=1}^{k/2} Y_i \ge \tfrac{k\mu}{2}(1+\delta)) \\
&\le \Pr(\mathsf{Bin}(\tfrac{k\mu}{2}(1+\delta), \tfrac{1}{\mu}) \le \tfrac{k}{2}) \\
&\le \exp(-\tfrac{k\delta^2}{2(1+\delta)}) \qquad (19)
\end{aligned}$$

upon applying a Chernoff bound for Binomial random variable. Now set $\xi = 4\ln(n)/k$ and $\delta = \xi/2 + \sqrt{\xi + \xi^2/4}$. Substituting this δ into (19) yields

$$\Pr\left(\sum_{i=1}^{k/2} T_i \ge \tfrac{k\mu}{2}(1+2\delta)\right) \le \frac{1}{n^2} \qquad (20)$$

and so,

$$\begin{aligned}
\sum_{i=1}^{k/2} T_i &\le \frac{kq}{q-1}\left(1 + \tfrac{2\ln(n)}{k} + \sqrt{\tfrac{4\ln(n)}{k} + \tfrac{4(\ln(n))^2}{k^2}}\right) \\
&= \frac{q}{q-1}\left(k + 2\ln(n) + \sqrt{4k\ln(n)} + 4(\ln(n))^2\right) \\
&\le \frac{q}{q-1}(k + 4\ln(n) + 2\sqrt{k\ln(n)}) \quad w.p.\ 1 - O(\tfrac{1}{n^2})
\end{aligned}$$

□

A.3 Proof of Lemma 4.5

Lemma A.3. *For $k/2 < i < k - 14\sqrt{k\ln(k)}$, we have*

$$\mathbb{E}[T_i] \le \frac{2q}{q-1} + 1 + \log(\ln(2)) - \log\left(\ln(\tfrac{k'}{i})\right) + O(\tfrac{1}{n^2}),$$

where $k' = k - 14\sqrt{k\ln(k)}$. Further,

1. $\displaystyle\sum_{i=k/2}^{k-14\sqrt{k\ln(k)}} \mathbb{E}[T_i] \le \tfrac{k}{2}(\tfrac{2q}{q-1} + 1 + \log(\ln(2)) + \tfrac{1+\ln(2)}{\ln(2)}) + O(\sqrt{k\ln k})$

2. $\sum_{i=k/2}^{k-14\sqrt{k\ln(k)}} T_i \leq \frac{k}{2}(1+\log(\ln(2))+\frac{1+\ln(2)}{\ln(2)}+\frac{2q}{q-1})+O(\ln(n))+O(\sqrt{k\ln(n)}),\ \ w.p.\ 1-O(\frac{1}{n^2})$

Proof. Recall that $k' = k - 14\sqrt{k\ln(k)}$. Note that, $k/((1+\epsilon_n)(1+1/q)) > k'$, since

$$\begin{aligned}\frac{k}{(1+\epsilon_n)(1+1/q)} &= k - k(1-\tfrac{k}{(1+\epsilon_n)(1+1/q)}) \\ &\geq k - k(\epsilon_n + 1/q + \epsilon_n/q) \\ &\geq k - (7\sqrt{k\ln(k)} + 1 + \epsilon_n) \\ &\geq k' \,.\end{aligned}$$

Note from Lemma 4.2 that, T_i is the absorption time of the Markov chain given in Figure 5. Let $\mathcal{D}_i$ denote the event that the absorption does not happen in any of the states $\{1, 2, 3, \ldots, \overline{s}_i - 1\}$. It is clear that

$$\mathbb{E}[T_i] \leq \mathbb{E}[T_i|\mathcal{D}_i] = (1-\tfrac{1}{n^3})^{-1}\overline{s}_i + \mathbb{E}[\text{Geom}(\tfrac{1}{2}(1-\tfrac{1}{q}))]$$

since the absorption probability in the state $\overline{s}_i$ is $(1/2)(1-1/q)$. It follows that

$$\mathbb{E}[T_i] \leq (1-\tfrac{1}{n^3})^{-1}\overline{s}_i + \tfrac{2q}{q-1} \tag{21}$$

Note that, from Lemma 4.2,

$$\begin{aligned}\overline{s}_i &= \min\{s : (\tfrac{i}{k})^{2^s}\left((1+\tfrac{1}{q})(1+\epsilon_n)\right)^{2^s-1} \leq \tfrac{1}{2}\} \\ &\leq \min\{s : \left(\tfrac{i}{k}(1+\tfrac{1}{q})(1+\epsilon_n)\right)^{2^s} \leq \tfrac{(1+\epsilon_n)(1+1/q)}{2}\} \\ &\leq \min\{s : \left(\tfrac{i}{k'}\right)^{2^s} \leq \tfrac{(1+\epsilon_n)(1+1/q)}{2}\}\end{aligned}$$

since we have shown that $k/((1+\epsilon_n)(1+1/q)) > k'$. This yields

$$\overline{s}_i = \lceil \log\left(\ln(\tfrac{2}{(1+\frac{1}{q})(1+\epsilon_n)})\right) - \log\left(\ln(\tfrac{k'}{i})\right)\rceil \leq 1 + \log(\ln(2)) - \log\left(\ln(\tfrac{k'}{i})\right) \tag{22}$$

Now, using the inequality $\ln x > 1 - 1/x$,

$$\begin{aligned}
&\sum_{i=k/2}^{k'-1} -\log\ln(\tfrac{k'}{i}) \\
&< \tfrac{1}{\ln(2)} \textstyle\sum_{i=k/2}^{k'-1} -\ln(1-\tfrac{i}{k'}) \\
&= \tfrac{1}{\ln(2)} \textstyle\sum_{i=k/2}^{k'-1} \ln(\tfrac{k'}{k'-i}) \\
&< \tfrac{1}{\ln(2)} \ln(\tfrac{{k'}^{k'-k/2}}{(k'-k/2)!}) \\
&\leq \tfrac{1}{\ln(2)} \ln(\tfrac{{k'}^{k'-k/2}}{((k'-k/2)/e)^{k'-k/2}}) \\
&= \tfrac{k'-k/2}{\ln(2)}(1-\ln(1-\tfrac{k}{2k'})) \\
&= \tfrac{k'-k/2}{\ln(2)}(1+\ln(2)) + O(\sqrt{k\ln(k)}) \qquad (23)
\end{aligned}$$

The inequality (23) along with (22) implies

$$\sum_{i=k/2}^{k'-1} \overline{s}_i \leq \tfrac{k}{2}(1+\log(\ln(2)) + \tfrac{1+\ln(2)}{\ln(2)}) + O(\sqrt{k\ln k}). \qquad (24)$$

Part 1 of Lemma follows from (21) and (24). For the high probability calculation, note that (see Figure 5)

$$T_i < s_i + \text{Geom}(\tfrac{2q}{q-1}), \;\; w.p.\; (1-\tfrac{1}{n^3})^{\overline{s}_i} \,,$$

from which it follows that

$$\sum_{i=k/2}^{k'-1} T_i = \sum_{i=k/2}^{k'-1} \overline{s}_i + \sum_{i=k/2}^{k'-1} F_i, \;\; w.p.\; (1-\tfrac{1}{n^3})^{\sum_{i=k/2}^{k'-1}\overline{s}_i} = (1-\tfrac{1}{n^3})^{O(n)} \geq 1 - O(\tfrac{1}{n^2}) \qquad (25)$$

where $F_i's$ are iid and $F_i \sim \text{Geom}(2q/(q-1))$. Further, using the Chernoff bound for Geometric random variables as in the proof of Lemma 4.4, we also have

$$\sum_{i=k/2}^{k'-1} F_i \leq \frac{q}{q-1}(k + 4\ln(n) + 2\sqrt{k\ln(n)}), \; w.p.\; 1 - O(\tfrac{1}{n^2}), \qquad (26)$$

and thus it follows from (24), (25), and (26) that,

$$\sum_{i=k/2}^{k'-1} T_i = \tfrac{k}{2}(1+\log(\ln(2)) + \tfrac{1+\ln(2)}{\ln(2)} + \tfrac{2q}{q-1}) + O(\ln(n)) + O(\sqrt{k\ln(n)}), \;\; w.p.\; 1 - O(\tfrac{1}{n^2}) \qquad (27)$$

We have proved part 2 of Lemma. □

A.4 Proof of Lemma ??

Lemma A.4. *For $i > k - 14\sqrt{k\ln(k)}$, we have*

1. $\mathbb{E}T_i = O(\ln(n))$
2. $\sum_{i>k-14\sqrt{k\ln(k)}} T_i = O(\sqrt{k\ln(k)}(\ln(n)),\ \ w.p.\ 1 - O(\frac{1}{n^2})$.

Proof. Note that, it immediately follows from Lemma 4.3 that

$$\mathbb{E}[T_i] \leq \frac{\ln(n)}{1-\sqrt(0.4)} + \frac{\ln(n/2)}{0.4\ln(4/3)} + \frac{2q}{q-1} .$$

For the second part, simply note that,

$$\sum_{j=1}^{\ln(n)} G_j = O(\ln(n))\ \ w.p.\ 1 - O(\tfrac{1}{n^3}) ,$$

where G_k's are the geometric random variables given in Lemma 4.3, and similarly for the H_j's. Thus

$$T_i = O(\ln(n)) + \text{Geom}(\tfrac{1}{2}(1-\tfrac{1}{q}))\ \ w.p.\ 1 - O(\tfrac{1}{n^3}) .$$

The result follows upon further applying the Chernoff bound for the geometric random variables with parameter $(1/2)(1-1/q)$. □

A.5 Proof of Lemma 5.4

Lemma A.5. *For $i \geq k - \sqrt{k}\ln(k)$,*

$$T_i \prec_{st} \sum_{j=1}^{s_n} H_j + \textit{Geom}(1 - \sqrt{\tfrac{2}{5}}) ,$$

where $G_j's$ are iid with distribution Geom(0.5), $H_j's$ are iid with distribution Geometric(θ) for some $\theta \in (0,1)$ and for large enough n. Further $r_k = \log_{(1+\gamma)}(n)$ ($\gamma = 0.2$).

Proof. The proof is not much different from that of a single message with a *push* mechanism. We provide a sketch of the proof below which is slightly different from the proof of Lemma 4.3.

Suppose at a particular time t, node u has dimension i in the regime considered in this lemma. We call a node helpful if the subspace spanned by the code-vectors at the node does not lie in that at u, and unhelpful otherwise. Note that, since the node u does not have full rank, there is at least one node that can potentially help node u if they call node u. Starting with a worst case scenario as in the proof of Lemma 4.2, assume that all the other $n-2$ nodes are unhelpful to u. The system then proceeds in rounds. Again, we find the time

required until there are at least $n/2$ nodes that can potentially help u. Since u may increase its dimension before that happens, this time provides a bound on the time u spends at dimension i. We thus consider the following two phases in the evolution.

1. The first phase ends when there are at least $n/2$ helpful nodes. This details of this phase almost parallels the proof of Lemma 5.3 and so we simply point out the differences without providing the details. Say, after a few rounds there are r helpful nodes and u has not increased its dimension. We have $1 \leq r \leq n/2$ in this phase. Denote by h_u the fraction of unhelpful nodes. We have $h_u = (n-r)/n$. Also, denote by h_u^+ the fraction of unhelpful nodes after the current round of message exchange. Let $l_1, l_2, \ldots l_{n-r}$ be the unhelpful nodes, and $Z_{l_j},\ j = 1, 2 \ldots (n-r)$ be $0-1$ random variables so that $Z_{l_j} = 1$ iff the node l_j does not increase its dimension after the current round of message exchange. Thus, $h_u^+ = \sum_j Z_{l_j}/n$. Using an argument similar to that in Claim 5.1 and Claim 5.2, we have

$$\begin{aligned}
&\Pr(h_u^+ \geq (1 - \tfrac{r}{n})(\tfrac{2}{5})^{(1-h_u)}(1+\epsilon)) \\
&= \Pr(\sum_{j=1}^{n-r} Z_{l_j} \geq (n-r)(\tfrac{2}{5})^{1-h_u}(1+\epsilon)) \\
&\leq \exp(-\tfrac{\epsilon^2(n-r)}{4}(\tfrac{2}{5})^{1-h_u}) \\
&\quad \text{(using calculations similar to(15)with } (n-r) \text{ in place of} \tfrac{ni}{k},\ \text{and } (\tfrac{2}{5})^{1-h_u} \text{ in place of } \overline{p}_u(t)) \\
&\leq \exp(-\tfrac{\epsilon^2 n}{8}\sqrt{\tfrac{2}{5}}) \quad (\because\ r \leq n/2,\ 1-h_u \leq \tfrac{1}{2} \text{ in this regime})\ .
\end{aligned}$$

By choosing $\epsilon = \epsilon_n = \delta/\sqrt{n}$, we have

$$\Pr(h_u^+ \geq (1 - \tfrac{r}{n})(\tfrac{2}{5})^{1-h_u}(1+\epsilon_n)) \leq \exp(-\tfrac{\delta}{8}\sqrt{\tfrac{2}{5}}) \triangleq \theta(\delta) < 1$$

where we will choose δ appropriately soon. Again, repeating the straight-forward, but tedious computation as in the derivation of (16) in the proof of Claim 5.2, we can show that,

$$\begin{aligned}
&h_u^+ \leq (1 - \tfrac{r}{n})(\tfrac{2}{5})^{1-h_u}(1+\epsilon_n) \\
&\Rightarrow h_u^+ \leq 1 - (1+\gamma)\tfrac{r}{n} \quad (\gamma = 0.2)
\end{aligned}$$

for a suitable choice of δ. Thus the number of helpful nodes gets multiplied by a factor $(1+\gamma)$ in every round with a constant probability of at least $1 - \theta(\delta)$ (instead of probability $1 - O(1/n^3)$ since $\epsilon_n = \delta/\sqrt{n}$). Thus, the sum of another $\ln(n)/(\ln(1+\gamma))$ geometric random variables takes the system to the state when there are half the nodes that can potentially increase the dimension of node u.

2. Finally, using Lemma 5.1, once there are at least $n/2$ that can help u, u can increase its dimension with probability at least $1 - \sqrt{\frac{2}{5}}$.

The result follows by combining the three cases. □