Graham, M. A., Ganesh, A. J., & Piechocki, R. J. (2019). *Sparse random linear network coding for low latency allcast*. Paper presented at 57th Annual Allerton Conference on Communication, Control, and Computing, Monticello, United States.

Peer reviewed version

Sparse random linear network coding for low latency allcast

Mark A. Graham
School of Mathematics
University of Bristol
Bristol, United Kingdom
mark.graham@bristol.ac.uk

Ayalvadi Ganesh
School of Mathematics
University of Bristol
Bristol, United Kingdom
a.ganesh@bristol.ac.uk

Robert J. Piechocki
Faculty of Engineering
University of Bristol
Bristol, United Kingdom
r.j.piechocki@bristol.ac.uk

***Abstract*—Numerous applications require the sharing of data from each node on a network with every other node. In the case of Connected and Autonomous Vehicles (CAVs), it will be necessary for vehicles to update each other with their positions, manoeuvring intentions, and other telemetry data, despite shadowing caused by other vehicles. These applications require scalable, reliable, low latency communications, over challenging broadcast channels. In this article, we consider the allcast problem, of achieving multiple simultaneous network broadcasts, over a broadcast medium. We model slow fading using random graphs, and show that an allcast method based on sparse random linear network coding can achieve reliable allcast in a constant number of transmission rounds. We compare this with an uncoded baseline, which we show requires $O(\log(n))$ transmission rounds. We justify and compare our analysis with extensive simulations.**

***Index Terms*—Sparse RLNC, CAV, Allcast, V2V, gossip**

I. Introduction

Emergent CAV systems have great potential to improve road safety and reduce congestion, amongst other benefits, and rely heavily on sharing data between vehicles to achieve their aims. For example, if vehicles share data such as their positions, acceleration and braking, they may cooperate in their manoeuvring. This allows vehicles to form *platoons* or cooperate on lane changing: making safer, more efficient use of the road network and saving both time and fuel [1]. Rather than point to point or broadcast links, these systems require a decentralised, distributed system for sharing messages amongst a group of nodes [2]. Every node on a given network has a message (or stream of messages) to share, simultaneously, and every node wishes to receive every one of these messages, a form of communication known as *allcast*.

The wireless channels between CAVs are notoriously harsh. In particular, the movement of vehicles through terrain causes unpredictable loss of communication links due to shadowing, notably as a result of obstructions caused by other vehicles [3]. Whilst modelling links between vehicles as erasure channels well models fast fading characteristics (such as multipath fading), this is not a good model of this form of slow fading, in which communication is not possible between nodes for much longer periods. In order to achieve low latency communications (if the fading will take longer to clear than the application may tolerate), the nodes must in some way cooperate, in order to aid the flow of each others data across the network.

In this paper, we compare the performance of a coded and an uncoded allcast system, for a system of nodes which are not all within communication range of one another. Our model will ignore fast fading, assuming that errors in individual communication links may be overcome using other methods (such as Forward Error Correction (FEC)).

II. Related work

Random Linear Network Coding (RLNC) is a well known method, in which coded messages are formed by taking linear combinations of message packets, with coefficients chosen at random from a finite field. Once each receiver has received as many linearly independent coded messages as there are message packets, the original messages may be decoded using Gaussian elimination. The application of RLNC to allcast was first proposed in [4], in the form of a *gossip algorithm*. The authors in this case adopt a *random phone call* model, in which users in each round select another single node, and transmit a coded message to them. They show that their method allows faster dissemination than uncoded methods. The work of [5] analyses RLNC gossip in great generality. Whilst their results are applicable to our model, the resulting bounds are not tight enough to be useful.

The authors of [6] consider allcast over complete undirected graphs, where the capacity of each edge is chosen i.i.d at random. The authors analyse the capacity region of their model, and present an uncoded push-pull allcast method, which they show to be asymptotically optimal. The authors do not however consider broadcast channels, and assume that different messages may be sent to each adjacent node (as they are modelling wired networks).

In [7], an RLNC allcast system is analysed, on graphs where edges denote erasure channels, and the medium is broadcast in nature. A multiple access system is in operation, restricting nodes to broadcast one at a time, and ensuring that

Research supported by the EPSRC grant EP/I028153/1, and Roke Manor Research.

nodes gain channel access with equal probability (modelling CSMA/CA). The authors show that the average stopping time (and total number of transmissions) for the complete graph is $O(n)$.

One barrier to the application of RLNC is the computational complexity involved in encoding and decoding the messages. One approach to mitigating this is to produce *sparse* random linear combinations, where coefficients are chosen to be 0 with some fixed probability π, and the remaining elements of the field are chosen with equal probability (i.e $\frac{1-\pi}{q}$, where q is the order of the finite field). This clearly reduces the encoding complexity (as fewer messages are included in each linear combination), and can reduce the computational expense of the Gaussian elimination algorithm [8]. RLNC codes with variable sparsity are analysed in [9], and a modification to the Gaussian elimination algorithm is presented, which allows lower complexity decoding of their codes. This method may provide some additional complexity reduction when used to decode our code. The use of sparse RLNC codes for broadcast transmissions over erasure channels is considered in [10]. The authors provide an accurate approximation for the probability of all users being able to decode every message.

III. Model

To model slow fading, we consider each transceiver to be a node on a random digraph $G = (V, E)$, $|V| = n$, $E \subseteq V \times V$, which is realised before communications commence, and remains constant for their duration. An edge exists from one node to another with probability p, and error and delay free communication between one node and another is possible exactly when an edge exists connecting them in that direction (in this way, our model makes no assumption of channel reciprocity), at any transmission opportunity. As we are modelling a broadcast channel, we assume that when a given node transmits a message, it transmits the same message to all of its neighbours.

We assume that every node broadcasts a message simultaneously, in synchronised "rounds", and that no interference between communications occurs. We further assume that no feedback or other control communications are possible, that the network topology is unknown to all transceivers, and that each transceiver has a buffer large enough to store every user's (decoded) message until the allcast is complete.

IV. Random message forwarding

As a baseline solution, we first consider a simple, uncoded method. To the best of our knowledge, this method first appeared in [4], named RMS (Random Message Selection).

In the first transmission round, as each node has so far received no data from the others, they can do no better (in any method) than to each broadcast their own packet to every other adjacent node; the alternative would be for some nodes not to use their first transmission round. So in every method, each node will initially broadcast its own packet.

Once each transmission round is complete, each node will add each packet received in that round which it had not previously received to its buffer (which is initialised with its own packet). In subsequent rounds, each node selects a packet from its buffer uniformly at random, and broadcasts it to its neighbours. This is repeated either for a fixed number of rounds (obviating what would be an expensive and impractical feedback system), or until all nodes have received all messages (a design choice).

Note that after the first round, the only way in which messages can be disseminated to more users (except in the rare case when G is the complete graph, in which a single round is sufficient) is if nodes further share messages from their buffers with other nodes. Without an expensive system of polling neighbours in order to learn the contents of their buffers, there is no way of knowing which packets are required by adjacent nodes, and it is inevitable that some relay transmissions will not be useful; this is the motivation for randomising the packet selection at each round.

We begin by stating the following lemma about the diameter of G: the least $d \in \mathbb{N}$ such that for each $i, j \in V$, there exists a path of length d or less from i to j.

Lemma 1. *Let d denote the diameter of G. Then $\mathbb{P}(d > 2) \leq \frac{n^2-n}{2}\left(1-p^2\right)^{n-2}$ and $\mathbb{P}(d < 2) \leq p^{\frac{1}{2}n(n-1)}$.*

Remark. By the above lemma, G will have diameter 2 with high probability as $n \longrightarrow \infty$.

Next, we define some notation which will be useful in the rest of the paper.

Definition 1. For each $i \in V$, we define the *in neighbourhood* $N_i^{\text{in}} = \{j \in V : (j, i) \in E)\}$, and refer to its members as *in neighbours* of i. Similarly, for each $i \in V$, we define the *out neighbourhood* $N_i^{\text{out}} = \{j \in V : (i, j) \in E\}$, and refer to its members as *out neighbours*.

We next recall a standard result about large deviations of binomial random variables, which is an immediate consequence of Sanov's theorem.

Lemma 2. *Suppose that X is a binomially distributed random variable with parameters (n, p), which we denote by $X \sim Bin(n, p)$. Then,*

$$\begin{cases} \mathbb{P}(X > nq) \leq \exp\big(-nH(q;p)\big), & \forall\, q > p, \\ \mathbb{P}(X < nq) \leq \exp\big(-nH(q;p)\big), & \forall\, q < p, \end{cases} \tag{1}$$

where

$$H(\beta;\alpha) = \beta \log \frac{\beta}{\alpha} + (1-\beta)\log\frac{1-\beta}{1-\alpha}$$

denotes the relative entropy or Kullback-Leibler (KL) divergence of the Bernoulli(β) distribution with respect to the Bernoulli(α) distribution.

We next prove a lemma which will be used to prove Theorem 1.

Lemma 3. *Let $g : [0,1] \longrightarrow [-1,1]$; $g(\lambda) = 1 - 2\lambda + \lambda\log(\lambda)$. Let $p \in (0,1)$, $n \in \mathbb{N}$, let $t = \frac{\alpha(1+\epsilon)}{p}\log(n)$, $\alpha >$*

0, $\epsilon > 0$, *let* $d = g^{-1}(p)(1-\epsilon)np\Big(1-\big(1-\frac{1}{n}\big)^t\Big)$. *Let* $X \sim$ *Bin*$\Big((1-\epsilon)np, 1-\big(1-\frac{1}{n}\big)^t\Big)$. *Then*

$$\mathbb{P}(X < d) < n^{-\alpha(1+\epsilon)}.$$

Remark. It is easy to show that $g(0) = 1$, $g(1) = -1$, and $g'(\lambda) \leq -1$. And since g is also continuous, g is invertible. Although no inverse exists in closed form, g^{-1} may be approximated numerically for practical purposes (its value is of no importance in the following outline proof).

Proof. We use Lemma 2 to bound $P(X > d) < e^{-f(n)}$, where $f(n) = nH\Big(\lambda\big(1-\big(1-\frac{1}{n}\big)^t\big); 1-\big(1-\frac{1}{n}\big)^t\Big)$. By expanding our expression for f, and using the standard bounds $\log(x) \leq x-1$ and $\big(1-(1-\frac{1}{n})^t\big) < \frac{t}{n}$ we may obtain

$$f(n) \geq \alpha(1+\epsilon)\log(n),$$

hence result. □

Theorem 1. *Consider a random graph $G = (V, E)$, where for each $e \in V \times V$, $\mathbb{P}(e \in E) = p \in (0,1)$, on which each node wishes to communicate a single message to every other node. Suppose the nodes implement the random forwarding approach: each node broadcasts its own packet, followed by a randomly selected packet from its buffer of packets it has received so far in each subsequent timestep. Let X denote the random number of timesteps before every node has received every message. Then*

$$\lim_{n\longrightarrow\infty} \mathbb{P}\left(X > \left(1+\epsilon+\frac{2(1+\epsilon)}{(1-\epsilon)\big(1-(1-p)^d\big)}\right)\frac{\log(n)}{p}\right) = 0.$$

Remark. Note that in contrast to [4], for our model, this method performs considerably better than a sequential store and forward approach. It is easy to show that such a method would require $2n$ transmission rounds with high probability, as a result of Lemma 1.

Proof. For $i \in V$, let $T_i^{(d)}$ be the time at which the d^{th} neighbour of i to have broadcast $i's$ packet first does so. By Lemma 2, and the union bound, we have $P(\cup_{i\in V}\big\{|n_i^{\text{out}}| < (1-\epsilon)np\big\}) \leq ne^{-\beta n}$, $\beta \in \mathbb{R}^+$. Each neighbour of a node i has at most n buffered packets, and the probability of each neighbour transmitting message i by time t is therefore greater than $1-(1-\frac{1}{n})^t$. Let $t = \frac{1+\epsilon}{p}\log(n)$, let $d = \Big\lfloor g^-1(p)(1-\epsilon)np\Big(1-\big(1-\frac{1}{n}\big)^t\Big)\Big\rfloor$, and notice by Lemma 3, that $\mathbb{P}(\cup_{i\in V}\big\{T_i^{(d)} \leq t\big\} > t) \leq n^{-\epsilon}$.

Assuming at least d nodes have broadcast each message by time t, we now show that a subsequent $t_1 = \frac{2(1+\epsilon)}{(1-\epsilon)\big(1-(1-p)^d\big)}\frac{\log(n)}{p}$ rounds are sufficient. Fix nodes $i, j \in V$. Each in in neighbour of j will will have received message i by time t if it is adjacent to one of the first neighbours of i to broadcast the message. Hence, the probability of a node being a neighbour of j and possessing packet i occurs with probability greater than $\big(1-(1-p)^d\big)p$. By Lemma 2, the number of such intermediate nodes is at least $(1-\epsilon)np\big(1-(1-p)^d\big)$ with probability at least $1-e^{-\gamma}$, $\gamma \in \mathbb{R}^+$. And since each packet is broadcast by one of these nodes with probability greater than $\frac{1}{n}$, the probability that t_1 subsequent rounds are not sufficient is at most

$$\big((1-\tfrac{1}{n})^{(1-\epsilon)np(1-(1-p)^d)}\big)^{t_1} \leq n^{-2(1+\epsilon)}.$$

Using the union bound to upper bound the union of these events over i, j yields the result.

□

V. Sparse random linear network coding

The random forwarding method is inefficient because the packet selected by a particular node for transmission in each round may not be useful to some of the nodes adjacent to it (if any), whilst the transmitting node may possess other packets which those less fortunate neighbours may not at that time. By coding across buffered packets, each node may communicate information about multiple packets in each round. This approach has the drawback that individual packets will not be decodable until all others are decodable, however for large networks the method has lower overall latency than the uncoded method regardless; the method trades this drawback for scalability and low latency on larger networks.

We now detail the sparse random linear network coded method. In the first round, as before, each node transmits its own packet (as it can do no better). Each node also buffers all these packets, keeping them separately from all subsequent packets, so that they, and they only, may be used to form coded packets for future transmission. Since by Lemma 1 the graph (with high probability) has diameter 2, if each node successfully communicates its packet and the contents of this buffer to all adjacent nodes, then every node will have decoded every message. Coding over packets which each node receives in subsequent rounds is possible (and may even be beneficial), but we define our method in this way for ease of analysis.

The set of nodes V is partitioned into a finite number of disjoint subsets S_j, each containing an equal number of elements $|S_1|$, except for the final set, which may contain extra elements if $|S_1|$ does not divide n evenly. This partition is globally known to all nodes, and decided before transmissions commence. Enumerate the following $\lfloor\frac{1}{p}\rfloor$ rounds by i. In each round i, each node broadcasts *partial* random linear combinations: random linear combinations of packets in the intersection of their buffer and S_i. Coefficients of each message are randomly chosen from $\mathbb{F}_2$, and are chosen to be 1 with probability π_1.

In subsequent rounds, each node broadcasts a *full* random linear combination of the messages in its buffer (with no restrictions/partitions), with coefficients chosen to be 1 with probability π_2.

Each node may decode the messages using Gaussian elimination once $n - |N^{\text{in}}| - 1$ linearly independent coded messsages have been received.

Whilst the code is rateless in the sense that every node will be able to decode all the messages if enough additional *full* random linear combinations are broadcast, in practice the feedback system necessary to determine when to stop would make this impractical, and instead the number of such rounds would be agreed in advance.

We assume that the coefficients are known to all users (by, for instance, using a pseudo-random number generator and sharing the seed amongst nodes), or that they may be communicated error free as part of a packet header.

We now begin by proving a series of lemmata, before proving the main result of this article. First, for each $i \in V$, we define M_i to be the matrix whose rows are formed by the coefficients of the linear combinations received by node i in the first $\lceil \frac{1}{p} \rceil$ rounds, including i's own packet and the messages received in the first round (which we consider to be trivial linear combinations, with exactly one non-zero coefficient). Note that since we are sampling from a single set S_j is each round, that the entries of each row of M_i will be zero in all columns except those which are members of exactly *one* set $|S_j|$.

Lemma 4. *Suppose V is partitioned into $\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil$ sets, with $|S_1| = \left\lceil \frac{n}{\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil} \right\rceil$. Then the probability that, for every j, the number of rows of M_i whose coefficients of members of S_j is at least $|S_j| - 1$, is at least $1 - ane^{-bn}$, $a, b \in \mathbb{R}^+$.*

Proof. A result of Lemma 2. □

Lemma 5. *Let δ_i denote the* defect *of M_i, the difference between its rank and n. If $\pi_1 = \frac{\log\left(|S_j| - (\lceil (1-\epsilon)p(|S_j|-1)\rceil + 1)\right)}{\left(|S_j| - (\lceil (1-\epsilon)^2 p(|S_j|-1)\rceil + 1)\right)p} = O\left(\frac{\log(n)}{n}\right)$, then $\mathbb{P}\left(\delta_i > n - \lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \rceil ((1+\epsilon)\log_2(n) + 1)\right) \leq n^{-(1+\epsilon)}$*

Proof. We assume that rows whose coefficients are non-zero in each set S_j are grouped together in blocks, that rows and columns are arranged so that each row contains an $|S_j| \times |S_j|$ square matrix of (possibly) non-zero coefficients, and that the first $\lceil (1-\epsilon)p|S_j| \rceil$ columns and rows contain an identity matrix. Note that we may eliminate the first $\lceil (1-\epsilon)p|S_j| \rceil$ columns of the remaining rows. There are at least $\lceil (1-\epsilon)p|S_j| \rceil$ neighbours in each set w.h.p by Lemma 2, and we assume that any packets in excess of this are discarded in the first round, and that any packets in excess of $|S_j| - 1$ in each block are discarded (note that there are at least this many by Lemma 4). Let δ_i^j denote the defect of this submatrix M_i^j in each block j. Discarding one column of each of these submatrices, we then apply Corollary 2.4 of [11] to each of the square submatrices remaining in each M_i^j, to obtain in each case

$$\begin{aligned}&\mathbb{P}(\delta_i^j > (1+\epsilon)\log(n) + 1) \\ &\quad < \mathbb{P}\left(\delta_i^j > (1+\epsilon)\log(|S_j| - \lceil (1+\epsilon)p|S_j| \rceil)\right) < n^{-(1+\epsilon)}.\end{aligned}$$

The result follows by applying the union bound over j. □

Theorem 2. *Suppose the nodes employ the coding scheme detailed above, with a single* full *transmission round (i.e $\lceil \frac{1}{p} \rceil + 1$ transmission rounds), and $\pi_2 = \frac{\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil \left((1+\epsilon)\log_2(n)+1\right)}{(1-\epsilon)np^2} = O\left(\frac{\log(n)}{n}\right)$. Let X denote the event that all nodes may decode every message once transmissions are complete. Then*

$$\lim_{n \longrightarrow \infty} \mathbb{P}(X^{\mathsf{c}}) = 0.$$

Remark. The reader may verify that the Theorem also holds for larger probabilities π_1', π_2', so long as $\pi_1 \leq \pi_1' \leq 0.5$ and $\pi_2 \leq \pi_2' \leq 0.5$. The case for π_1 is trivial, and the case for π_2 follows since the expected rank of each matrix M_i^j is monotone increasing for probabilities p_i' in this interval [11], meaning Corollary 2.4 of [11] and Lemma 5 of this article also hold.

Remark. Note that as a result of Lemma 2, each node will have at most $(1+\epsilon)(n-1)p$ in neighbours with high probability. As a result, at least $\frac{1}{(1-\epsilon)p}$ transmission rounds will be required by any method, as this is the minimum required for each node to receive $n-1$ messages in total. Hence, our method is close to optimal.

Proof. Fix $i \in V$. By Lemma 4 and Lemma 5, by the final transmission round, i will have received at least $n - \left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil ((1+\epsilon)\log_2(n)+1)$ linearly independent packets. Node i may decode all messages if and only if enough rows corresponding to the full linear combinations can be added to M_i to make it full rank.

In this case, we may reduce M_i to a matrix $\check{M}_i$, containing $n - \left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil \left((1+\epsilon)\log_2(n) + 1\right)$ linearly independent rows, with an identity matrix in the first $n - \left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil ((1+\epsilon)\log_2(n) + 1)$ rows and columns, and with all zero rows afterwards. We may then search the set of dense packets for ones which are linearly independent with the non-zero rows already in $\check{M}_i$, discarding those that are not, and adding those that are to $\check{M}_i$, re-arranging the matrix in the same way each time. If we are able to add $\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil \left((1+\epsilon)\log_2(n) + 1\right)$ rows to $\check{M}_i^j$ in this way, then the system is full rank.

Let J_i be the set of nodes corresponding to non-identity columns in $\check{M}_i$. Following the proof of Theorem 6.3 in [11], we notice that a dense packet is linearly dependent with the existing rows of $\check{M}_i$ if and only if it is a member of the subspace spanning its rows. If we choose the first $n - \delta_i$ columns arbitrarily, the final δ_i columns are uniquely determined by them; differing in any one of the final δ_i columns then implies linear independence. Notice that an element of a dense row may only be equal to 1 if the node which sent it is adjacent to the corresponding node. For each of the determined columns, there are at least $(1-\epsilon)np^2$ such nodes (as by Lemma 2, $|\{k \in N_i^{\text{in}} : (j,k) \in E\}| \geq (1-\epsilon)np^2$, w.h.p), but these in general will not be distinct for each column. If we limit our search to $(1-\epsilon)np^2$ packets, we can guarantee to find a packet which is adjacent to any column on each draw.

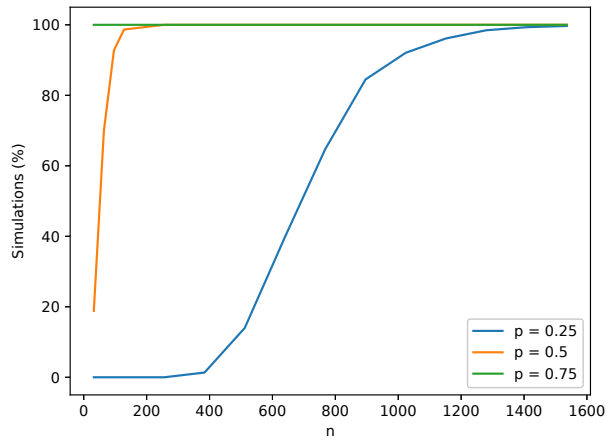

Fig. 1. Graphic showing the proportion of simulations of the network coded method, in which the number of rounds required was less than or equal to predicted values.

The probability that a given packet in this set differs in one particular column is at least π_2, and these probabilities are independent amongst packets and columns, as coefficients are chosen independently amongst nodes. We may now view successfully adding $\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil \Big((1+\epsilon)\log_2(n)+1 \Big)$ rows to $\check{M}_i$ in this way as achieving $\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil ((1+\epsilon)\log_2(n)+1)$ successes in $(1-\epsilon)np^2$ trials, i.e if

$$Y \sim \mathrm{Bin}\left((1-\epsilon)np^2, \frac{\left\lceil \frac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil \big((1+\epsilon)\log_2(n)+1 \big)}{(1-\epsilon)^2 np^2} \right)$$

then

$$\begin{aligned} \mathbb{P}(X_i^{\mathsf{c}}) &\leq \mathbb{P}\Big(Y < \left\lceil \tfrac{1-(1-\epsilon)p}{(1-\epsilon)p} \right\rceil \big((1+\epsilon)\log_2(n)+1 \big) \Big) \\ &\leq e^{-an} \end{aligned}$$

by Lemma 2, where $a \in \mathbb{R}$. Taking the union bound over i completes the proof. □

VI. Simulation results

In this section, we compare the asymptotic bounds from section V with extensive Monte Carlo simulation results, written in CUDA C. Figure 1 shows the proportion of 20000 simulations of the sparse RLNC system detailed in Section V, in which the system completed an allcast in $\left\lceil \frac{1}{p} \right\rceil + 1$ transmission rounds or fewer, for various edge probabilities p. Notice how quickly the system approaches predictions.

VII. Conclusion

In this paper, a method based on sparse random linear network coding was introduced, which can achieve allcast communications in a constant number of rounds. This method was compared to an uncoded baseline method, which requires $O(\log(n))$ transmission rounds to achieve the same aim, incurring an intollerable and inpractical amount of latency. We presented Monte Carlo simulations which showed the rapid convergence of our asymptotic bound on the number of transmissions required by the coded method.

VIII. Acknowledgements

The authors would like to thank Stephen Wales, Roke Manor Research, for helpful discussions.

References

[1] U. Montanaro, S. Dixit, S. Fallah, M. Dianati, A. Stevens, D. Oxtoby, and A. Mouzakitis, "Towards connected autonomous driving: review of use-cases," *Vehicle System Dynamics*, vol. 57, no. 6, pp. 779–814, 2019.

[2] F. Bai and B. Krishnamachari, "Exploiting the wisdom of the crowd: localized, distributed information-centric vanets [topics in automotive networking]," *IEEE Communications Magazine*, vol. 48, no. 5, 2010.

[3] M. Boban, T. T. Vinhoza, M. Ferreira, J. Barros, and O. K. Tonguz, "Impact of vehicles as obstacles in vehicular ad hoc networks," *IEEE journal on selected areas in communications*, vol. 29, no. 1, pp. 15–28, 2011.

[4] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2486–2507, 2006.

[5] B. Haeupler, "Analyzing network coding gossip made easy," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 2011, pp. 293–302.

[6] V. N. Swamy, S. Bhashyam, R. Sundaresan, and P. Viswanath, "An asymptotically optimal push–pull method for multicasting over a random network," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5075–5087, 2013.

[7] M. H. Firooz and S. Roy, "Data dissemination in wireless networks with network coding," *IEEE Communications Letters*, vol. 17, no. 5, pp. 944–947, 2013.

[8] M. V. Pedersen, D. E. Lucani, F. H. Fitzek, C. W. Sørensen, and A. S. Badr, "Network coding designs suited for the real world: What works, what doesn't, what's promising," in *2013 IEEE Information Theory Workshop (ITW)*. IEEE, 2013, pp. 1–5.

[9] S. Feizi, D. E. Lucani, C. W. Sørensen, A. Makhdoumi, and M. Médard, "Tunable sparse network coding for multicast networks," in *2014 International Symposium on Network Coding (NetCod)*. IEEE, 2014, pp. 1–6.

[10] S. Brown, O. Johnson, and A. Tassi, "Reliability of broadcast communications under sparse random linear network coding," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4677–4682, 2018.

[11] J. Blömer, R. Karp, and E. Welzl, "The rank of sparse random matrices over finite fields," *Random Structures & Algorithms*, vol. 10, no. 4, pp. 407–419, 1997.