On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks

Xiaolan Zhang, Giovanni Neglia[§] ,Jim Kurose, Don Towsley
University of Massachusetts [§]University degli Studi di Palermo
{ellenz,kurose,towsley@cs.umass.edu} giovanni.neglia@tti.unipa.it

February 15, 2006

Abstract

In this paper, we investigate the benefits of using a form of network coding known as Random Linear Coding (RLC) for unicast communications in a mobile Disruption Tolerant Network (DTN) under epidemic routing. Under RLC, DTN nodes store and then forward random linear combinations of packets as they encounter other DTN nodes. We first consider RLC applied to a single block of K packets where *(a)* all K packets have the same source and destination, *(b)* the K packets have different sources but a common destination and *(c)* the K packets each have a different source/destination pair; we also consider the case where blocks of K packets arrive according to a Poisson bulk arrival process. Our performance metric of interest is the delay until the last packet in a block is delivered. We show that for the single block case, when bandwidth is constrained, applying RLC over packets destined to the same node achieves (with high probability) the minimum delay to deliver the block of data. We find through simulation that the benefit over non-network-coded packet forwarding increases further when buffer space within DTN nodes is limited. For the case of multiple blocks, our simulations show that RLC offers only slight improvement over the non-coded scenario when only bandwidth is constrained, but more significant benefits when both bandwidth and buffers are constrained. We remark that when the network is relatively loaded, RLC achieves improvements over non-coding scheme only if the spreading of the information is appropriately controlled.

1 Introduction

Epidemic routing ([15, 13, 14, 9, 19]) has been proposed for routing in mobile disruption tolerant networks (DTNs) in which there may not be a contemporaneous path from source to destination. Epidemic routing adopts a so-called "store-carry-forward" paradigm – a node receiving a packet buffers and carries that packet as it moves, passing the packet on to new nodes that it encounters. Analogous to the spread of infectious diseases, each time a packet-carrying node encounters a new node that does not have a copy of that packet, the carrier is said to *infect* this new node by passing on a packet copy; newly infected nodes, in turn, behave similarly. The destination receives the packet when it first meets an infected node.

Random Linear Coding (RLC) is a form of network coding [2] where each network node, rather than forwarding packets unchanged along the path from source-to-destination, can forward random linear combination of the data it has received. The short primer [7] gives a nice review on the past work on network coding.

In this paper, we investigate the use of RLC in epidemic routing for unicast applications in mobile DTNs through simulation. In this case there are different possible ways to combine packets: each nodes can combine all the packets in its buffer, or only the packets destined to the same destination, or only the packets belonging to the same flow (i.e., same source-destination pair). We first consider these three possibilities in the simpler case of a single block of K packets. We then consider the case where blocks of K packets arrive according to a Poisson bulk arrival process. Our performance metric of interest is the delay until the last packet in a block is delivered. We show that for the single block case, when bandwidth is constrained, applying RLC over packets destined to the same node achieves (with high probability) the minimum delay to deliver a block of data. We find through simulation that this benefit increases further when buffer space within the DTN nodes is limited. For the case of multiple source/destination pairs, our simulations show that RLC offers only slight improvement over the non-coded scenario when only bandwidth is constrained, but more significant benefits when both bandwidth and buffers are constrained.

Several previous research efforts have applied *source-based* (i.e., non-network-coded) erasure codes to DTNs. [16] proposes erasure-coding-based routing for *opportunistic networks*, where DTN nodes operate without prior knowledge of node mobility patterns. For the case that a DTN has prior knowledge about paths and their loss behavior, [11] considers how to allocate the source-erasure-coded blocks to these paths. [6] proposes the usage of decentralized erasure codes to spread and store distributed data in a large scale sensor network. We note that this is similar to [16], in the sense that original packets traverse two hops to the final destination.

RLC has also been previously applied to networked scenarios including P2P content distribution ([8]), multicast application([3]), gossip protocol ([5, 4]) and distributed storage ([4, 1]). To our knowledge, the only work applying network coding in a DTN is [17], where the authors consider *broadcast* data delivery using RLC; our focus here is on using RLC for unicast delivery in a DTN.

The remainder of this paper is structured as follows. We introduce the network model and the forwarding schemes in Section 2. The simulation setting is described in Section 2. Section 4 studies the benefit of RLC over non-coded scheme for the scenario where there is a single generation of packets in the network. Section 5 extends the study to multiple generation case. Finally, Section 6 summarizes the paper and discusses future work.

2 Network model and Forwarding Schemes

We consider unicast communications (i.e. each messages has a single node as destination) in a network consisting of N nodes moving according to a mobility model (discussed shortly) within a closed region. Each node has a fixed limited transmission range, such that the network is sparse and therefore disconnected. When two nodes come within transmission range of each other (i.e., they meet), they first figure out if the other has some useful information and, if any, they try to exchange it. We detail this process with reference to the two mechanisms we are going to compare: traditional non-coded packet-forwarding and RLC-forwarding.

Non-coded forwarding: When two nodes meet, each of them randomly selects one or more packets, depending on the bandwidth, among the packets that the other node does not have, and forwards them to the other node. We refer to this as the **random** selection scheme. We also consider a **RR_random** scheme in which the packet's source node chooses a packet to forward in round-robin manner, while intermediate nodes use random selection. Our intuition is that RR selection will help to speed up the propagation of initial copies of each packet.

Random Linear Coding based forwarding: RLC is applied to a finite set of K packets, called *generation*. Each packet is viewed as a d dimensional vector over a finite field, F_q of size q. We denote by $m_i \in F_q^d, i = 1, 2, ...K$ the K packets. A linear combination of the K packets is:

$$f_l = \sum_{i=1}^{K} \alpha_i m_i, \alpha_i \in F_q.$$

Addition and multiplication are over F_q. Initially, the source node(s) carries the original packets (a linear combination with special coefficients $\alpha_i = 1, \alpha_j = 0, j \neq i$). If a node carries r independent linear combinations, we say that the rank of the node is r, and refer to the Kxr matrix made up of the coefficients of the r combinations as the node's encoding matrix.

We consider RLC scheme where signaling is used to first determine if two nodes have useful information to each other. When two nodes meet, the signalling process involves exchange of their encoding matrices. Each node, based on the matrix, checks if it has useful information to the other node. In fact, if a node has at least one combination that cannot be linearly expressed by the commbinations stored in another node, it has useful (innovative) information for the latter node. If so, the node generates a random linear combination of the currently stored combinations, say $f_1, ...f_r$, by selecting uniformly at random the coefficients $\beta_1, ...\beta_r$ over the field F_q, and generates: $f_{new} = \sum_{j=1}^{r} \beta_j f_j$. Obviously, f_{new} is a linear combination of the original K packets. This new combination, along with the coefficients in terms of the original packets, is forwarded to the other node. Note that this combination is useful to the other node with probability no less than $1 - 1/q$ according to the Lemma 2.1 in [5]. [1] When a node (e.g., the destination) reaches rank K (i.e., full rank), it can decode the original K packets through matrix inversion[2]. Notice that RLC based scheme incurs storage overhead for storing coefficients for each combinations and requires more computation to check if one node has useful information for the other and for the receiver to decode the original packets through performing matrix inversion. We will discuss the overhead of RLC scheme in Section **??**.

As each packet is duplicated by the nodes in the network, when it is first delivered to the destination, there are multiple copies of the packet in the network. A recovery scheme can be used to delete these obsolete copies from the network [10] to free up storage space and save avoid useless transmission. We will use VACCINE recovery scheme throughout this paper. Under VACCINE, when a packet is first delivered, an *antipacket* is generated and propagated through the network to delete buffered copies of this packet. Under RLC scheme, when a generation is delivered to the destination, an antipacket for the generation is generated and propagated to delete buffered combinations of the generation. To simplify analysis and simulation, we assume that the storage and transmission of antipacket is not subject to bandwidth and buffer constraints.

We study the time to deliver a block of K packets when the packets are forwarded without any coding or when RLC is applied to the block of K packets. In particular, we define *block delivery delay* as the time from the arrival of the block in the network to the delivery of the whole block to the destination, denoted by D_{block}. Depending on the specific application, other metrics could be more meaningful, like the average time to deliver a packet of the block, or the average time to deliver a packet respecting the order. Note that D_{block} is the metric more favorable to RLC in the comparison. Another performance metric of interest is the

[1] Note that a node, with the encoding matrix of the receiving node, can iterate this process of generating random linear coefficients until generating a combination that is useful to the receiving node.

[2] A packet can be decoded before the matrix reachs full rank, as long as the encoding matrix contains a simple encoding coefficient.

average number of packet copies or combinations made within the network, as this is a measure of resources consumed (bandwidth, transmission power, buffering) within the DTN.

3 Simulation Setting

We perform our simulation study using our own simulator. Rather than simulating a specific mobility model (e.g., the random waypoint or random direction mobility model), we directly simulate a pair-wise Poisson meeting process between two nodes. [9] has shown that under the random waypoint/direction models, the inter-meeting time between a pair of nodes follows a Poisson process when node velocity is relatively high compared to the region size, and the transmission range is relatively small. This simplification speeds up the simulation. We have also performed simulations using the actual mobility models and observe similar performance. Due to space constraints, these latter results are not presented here. For the results presented in this paper, we simulate a network of $N = 101$ nodes with a pair-wise meeting rate of $\beta = 0.0049$. We use a finite field of size $q = 701$.

4 Single Generation Case

In this section, we focus on the simple setting where there is a single generation of packets in the network. In particular we assume that K packets arrive at the same time in the network (we will discuss the application of RLC scheme when packets arrive individually in Section **??**). We examine the following three scenarios:

- **SS_SD (Single Source/Single Destination):** in which data in K packets from a source are to be delivered to a single destination;
- **MS_SD (Multiple Source/Single Destination):** in which data in K packets from different sources are to be delivered to the same destination;
- **MS_MD (Multiple Source/Multiple Destination):** in which data in each of K packets (each from a different source) are to be delivered to a different destination.

4.1 Benefit of coding under bandwidth constraints

We first consider the case when bandwidth is constrained, i.e., when two nodes meet, they can send a maximum of b packets in each direction. We assume for now that mobile nodes have sufficient buffer space to store all packets.

Claim 1 *If there is a single block of packets in the network, for the SS_SD and MS_SD case, RLC achieves the minimum D_{max} with high probability.*

We provide an intuitive argument making use a random multigraph (a temporal network) constructed as follows (Fig.1): there are N vertices, each corresponding to one mobile node. For each contact between a pair of nodes that can exchange b packets in each direction, b directed edges are added in each direction between the corresponding vertices. Edges are labeled with the time that the contact occurs. A *time-respecting*

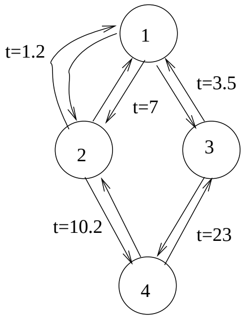


Figure 1: Random graph representing the contacts between nodes

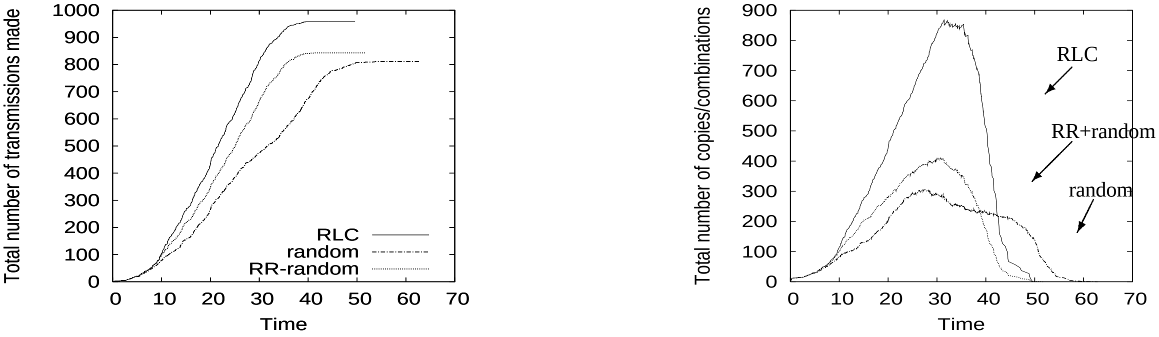
path in such a network is a path in the graph where the successive edges have increasing timestamps. A set of paths are *independent* if they do not share edges. Previous work [12] has studied connectivity and inference problems for such *temporal networks* that arised from applications such as communication in distributed networks and epidemiology etc. Our model captures one more aspect of the network: the capacity of the contact.

For the SS_SD case, where the source node initially has K packets to send at $t = 0$, the time to deliver these K packets cannot be smaller than the time when there are K independent paths from the source to the destination. Similarly, for the MS_SD case, the delivery time cannot be smaller than the time to have K independent paths from the K source nodes to the destination. For non-coding scheme, this minimum delay is hard to achieve. As each node has no knowledge about packets transfers happening among other nodes, it is likely that the nodes along some of these K paths forward packets that other paths are propagating. **Under RLC scheme, rather than choosing from the K packets, nodes randomly and independently encode packets to generate "equally important" encode-packets. As the number of independent coded packets is much greater than K, the probability that some path select to forward a packet(information) that is useless to the destination is much smaller than non-coded scheme.**

We use an example of 4-node network as shown in Fig.1 to illustrates this idea. Assume that node 1 generates two packets m_1, m_2 destined to node 4 at $t = 0$. Given the contact scenario shown in the figure, at time $t = 23$, there are two edge-disjoint paths from node 0 to node 4; therefore the minimum delay to deliver the two packets is 22. Without applying network coding, node 1 forwards m_1, m_2 to node 2 at time $t = 1.2, 7$ respectively, and one of the packets (say m_1) to node 3 at time $t = 3.5$. When nodes 2 and 4 meet at $t = 10.2$, node 2 randomly selects a packet and delivers to node 4 (as it has no global knowledge of past and future contacts for other nodes). With probability 0.5, packet m_2 is selected to forward to node 4, and thus when node 3 meets node 4 at $t = 23$, it has no useful information for node 4. For this particular example, non-codded scheme achieves minimum delay with probability 0.5. On the other hand, under RLC scheme, source node 1 forwards random linear combination c_1, c_2 to node 2, and c_3 to node 3 at the contact at time $t = 1.2, 7, 3.5$ respectively. With proability $1 - 1/q$, c_1, c_2 are independent. If c_1, c_2 are independent, node 2 stores both combinations. When node 2 meets node 4, it generates a random linear combination c_{12} of c_1, c_2 and forwards it. If c_{12} and c_3 are independent, node 4 can decode the two original packets after node 3 delivers c_3 at time $t = 23$. Note that c_3 can be linearly expressed by c_1, c_2, and with probability $1 - 1/q$, c_{12} is independent from c_3. For the case where c_1, c_2 are linearly dependent, node 2 stores one of them, say c_1, and forwards it to node 4 at $t = 10.2$. If c_3, c_1 are independent (with probability $1 - 1/q$), node 4 reaches full rank at $t = 23$ and the two packets are delivered at the minimum delay. So, for this particular network and contact process, RLC achieves minimum block delivery delay with probability $1 - 1/q$. Characterizing this probability for arbitrary large network and meeting process becomes infeasible due to the explosion of the number of possible situations.

As [7] observed, the benefits of network coding in problems such as gossip ([5]), P2P content distribution ([8]) and broadcast application in wireless ad hoc network ([17, 18]) stems from its robustness and adaptibility that can be captured by the generic Coupon Collector Problem. For the same reason, in our settings, RLC based scheme can achieve the minimum delay to deliver a single block of packets with high probability. However, unlike the problems previously studied which are basically broadcast applicaitons, we are concerned with unicast applications. For unicast applications, there is a tradeoff between delay and resource consumption (i.e.,transmission power and buffer space). Therefore the ultimate benefit of network coding is less evident. We will explore these tradeoff later in Section 4.3.

We first highlight several characteristics of RLC compared to a non-coded approach using simulation result. For a particular run for SS_SD case with $N = 101, K = 10$ case, Fig.2.(a) and (b) depicts the accumulative number of transmissions made, the total number of packet copies (for the non-coding scheme) or combinations (for RLC scheme) in the entire network as a function of time respectively. We observe that RLC allows faster propagation of the information in the network, but also incurs more copies being made in the network. There are two factors causing more transmissions made under RLC: first, RLC allows faster propagation of information, as the random combination of packets at each node allow two nodes that meet each other to have useful information to exchange; secondly, under RLC scheme the recovery process starts only when the whole generation is delivered (much later than under non-coding approach, where recovery process for individual packet starts immediately when it is delivered). Another point to make is concerning the performance metric. Throughout this paper, we mainly study the block delivery delay as the performance metric; there are alterative metrics such as mean packet delay, in-order delay. For multiple simulation runs of the above setting, Figure 3 plots the empirical CDF for different delay metrics achieved by RLC and RR-random scheme. It shows that although RLC is able to decreases the block delivery delay, it sacrifices performance metrics such as mean packet delay, in-order packet delay.



(a) RLC scheme achieves faster propagation

(b) storage comsumption

Figure 2: RLC vs non-coding scheme

As a quantitative analysis of delivery delay is difficult due to the random nature of the contacts, and the large size of the networks in which we are interested, we use simulation to quantify the performance gain of RLC scheme.

We next explore the relative benefit of RLC with respect to the non-coded case under varying bandwidth constraints. Fig.4(a) plots the average D_{block} for SS_SD with $K = 10$ under varying bandwidth constraints. (We note that the average D_{block} reported throughout Section 4 are the average value from 50 different simulation runs). The figure shows that RLC achieves lower D_{block} than both random and RR_random schemes.

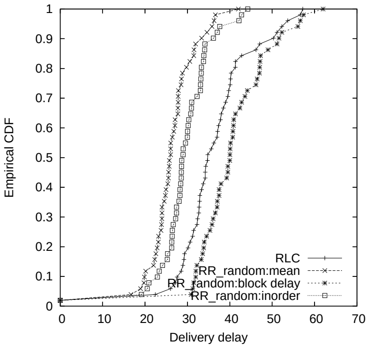
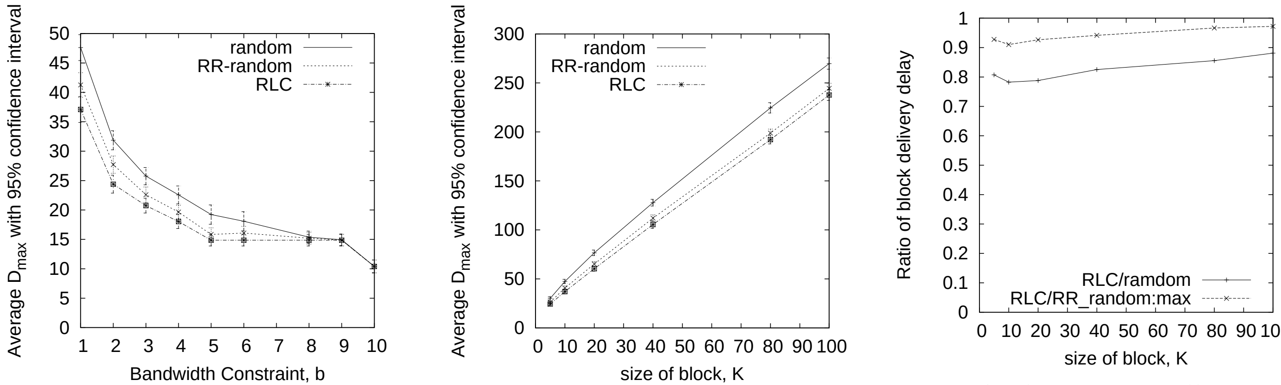


Figure 3: Trade-off between different performance metrics,$N = 101, K = 10$,SS_SD



(a) D_{max} under varying bandwidth

(b) D_{max} under different block size

(c) Benefit of RLC under different block size

Figure 4: RLC benefit under SS_SD

All schemes perform the same when $b = 10$ where the $K = 10$ packets are propagated independently without competing for bandwdith; whereas as bandwidth decreases, the relative benefit of RLC increases.

Fig.4(b) shows the sensitivity of performance to block size, plotting the average D_{max} for the SS_SD case with varying block size and a bandwidth constraint of $b = 1$ (i.e., on every contact, only one packet can be sent in each direction. For the remainder of this paper, this is the default bandwidth constraint used in our simulation results). From Fig.4(c), we observe that as the block size increases, the relative benefit of RLC over non-coding scheme decreases. This is because with larger block size, there are a larger number of packets to choose from, and therefore the probability of two paths choosing to forward the same packets is smaller.

Due to space constraint, our results for MS_SD and MS_MD case are not shown here. We note that the benefit achieved by RLC for the MS_SD case is smaller than for the SS_SD case. **redo this paragraph** This is because here the K packets start to propagate from K different nodes, the effect of relay nodes choosing the wrong packets to forward becomes less significant. For MS_MD case, RLC performs worse than the non-coding scheme since RLC forces every destination node to receive K independent combinations to decode the one single packet destined to it.

4.2 Benefit of Coding under Bandwidth and Buffer Constraints

Thus far, we have assumed that nodes have unlimited buffer capacity. In this section, we assume that the relay nodes can store at most $B(B < K)$ packets or combinations; source and destination nodes are not subject to this constraint. For RLC, when a node receives a combination and its buffer is full, it randomly combines the new combination with an existing combination in the buffer and stores the result. For the non-coding case, a drophead scheme ([19]) is used which drops the packet that has resided in the buffer the longest when a new packet arrives and the buffer is full.

Fig.6(a) shows that, for the SS_SD case (with $K = 10$), as nodal buffer sizes decrease, the performance of RLC degenerates only slightly; while the performance of the non-coding schemes degrade quickly. To explain the big benefit of RLC, we examine the simulation trace closer. For a particular run, Fig.5 plots the accumulative number of transmissions made as a function of time for different schemes. We see that RLC is able to make use of more transmission opportunities. **We find that under non-coding approach, different packets in the block propagate with very uneven speed: some packets spread quickly to a large number of nodes, while other packets propagates much slower. The uneveness of propagation can be explained by the adopted random selection scheme: the more copies a packet has in the network, the more likely it is fowarded to other node and kick out a copy of other packets when a carrier of the packet meets other nodes. RLC scheme does not suffer from such problem: different packets are mixed randomly by nodes, therefore, when a node drops a combination, equal amount of information is lost for each packet. As a result of such uneveness, it takes much longer to deliver the "slowest" packet in noncoded approach than in RLC scheme.**

Again, we note that the improvement in delay performance of RLC is achieved at the cost of more transmission made as shown in Fig.6(b). Notice that although under unconstrained buffer case, at most K linear combinations of a generation (of size K) are sent to each node, this is not the case under buffer constraint. When a relay node cannot store all combinations of a generation, it can be repeatedly sent different combinations of a generation without increasing rank.

For MS_SD case, we observe similar performance gains of RLC (not shown here). For MS_MD ($K = 10$) case where coding is applied to packets sent by different sources to different destinations, we observe that RLC out-performs non-coding random scheme when the buffer is very constrained ($K = 10, b = 1$ for this setting) as shown in Fig.6(c).

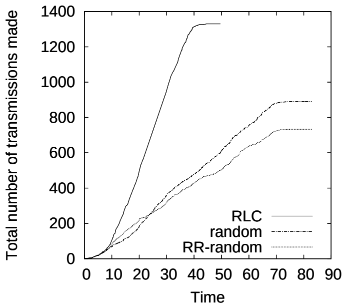
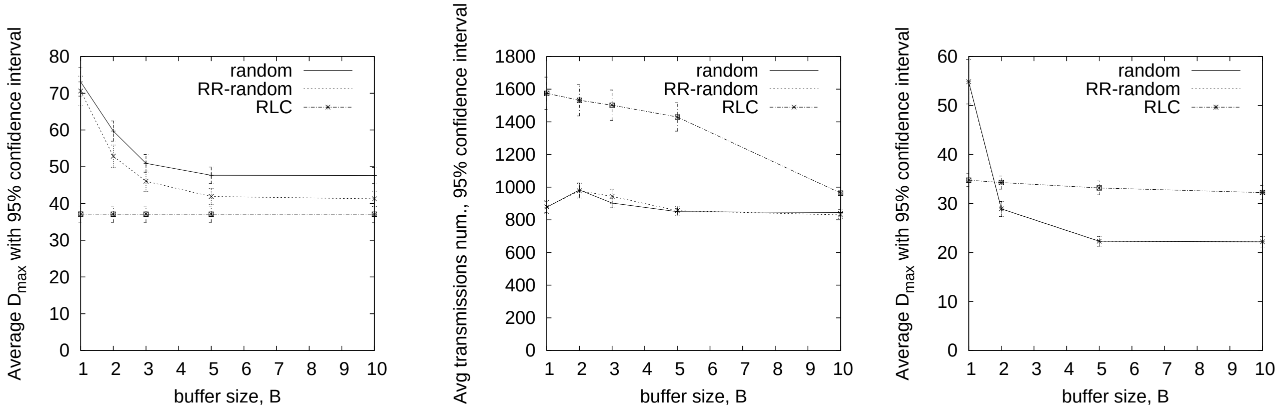


Figure 5: RLC makes use of more transmission opportunities,B=1



(a) D_{block} for SS_SD under varying buffer

(b) Number of transmissions made for SS_SD

(c) D_{block} for MS_MD under varying buffer size

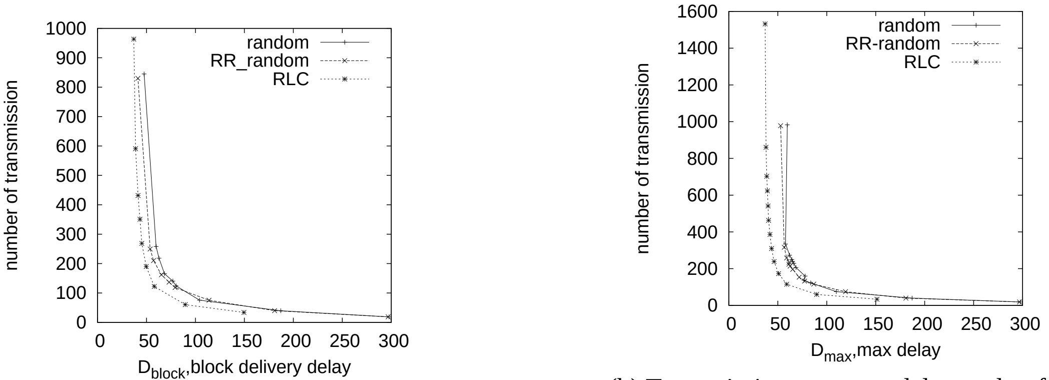
Figure 6: Bandwidth and buffer constraint

4.3 Controlling Transmission Power Consumption of RLC

We have seen that RLC is able to deliver a block of data, or collect multiple packets from different sources in the minimum amount of time, at the "cost" of having more copies of packets present in the network, consuming more buffer space, transmission power and bandwidth (to send these copies). Can RLC achieves smaller block delivery delay than non-coded scheme 1) under the same transmission power consumption, 2) under the same transmission power consumption and buffer constraint ? We address these questions in this section.

To limit the number of copies made for a packet, we use a token-based scheme, extending the *binary spray and wait* scheme proposed independently in [14, 13]. We refer to the maximum number of copies made for a packet as per-packet token number. Spray and wait protocol [14] with per-packet token L consists of two phases: spray phase to spread $L - 1$ copies of the packet, wait phase (if the destination node has not been reached) where each of L carriers (including the source) performs direct transmission to deliver the packet to the destination. There can be different ways to spread the initial $L - 1$ copies, one of them is binary spray and wait. Under binary spray and wait with per-packet token number L, every new packet generated at the source is assigned $L - 1$ tokens. When the source node meets another node, the packet is copied to the other node and half of the tokens are assigned to the new copy, while the source node keeps the remaining half of the tokens. A relay node carrying a copy in turn does the same. When a packet copy has only a single token remaining, it can only be forwarded to the destination. [14] shows that under independelty and identically distributed mobility model, binary spray and wait achieves minimum expected delay among all spray and wait routing. We note that this scheme can be improved by allowing two nodes carrying copies of the same packet to average their token numbers when they meet, as the two nodes have equal opportunities to meet susceptible nodes or destination node (and to propagate and deliver the packet).

We extend the notion of tokens to the RLC scheme by associating a token number with the generation, which equals to the product of the number of packets in the generation and the per-packet token number. The token number for a generation limits the total number of combinations that can be exchanged in the network. When a node sends a random combination to another node, its token number is decreased by one. After two nodes finish exchanging combinations for a generation, they average their tokens: the total of two nodes' token number is reallocated to the two nodes in propotion to their ranks. Even if the two nodes meet each other have no information to exchange, they average their token numbers. The rational of averaging



(a) Transmission power vs delay trade-off

(b) Transmission power vs delay trade-off under buffer constraint, B=2

Figure 7: Power vs delay trade-off achieved with different token number

tokens is that the potential of a node to spread the generation is linear with the rank of the node (i.e., the amount of information the node carries for the generation).

We run simulations for SS_SD case with $K = 10$ with per-packet token numbers varying between 5 and 90, and unlimited. Fig.7 plots ((a) without buffer constraint,(b) with buffer constraint of 1), the number of transmissions versus delay tradeoff achieved under different per-packet token limits.

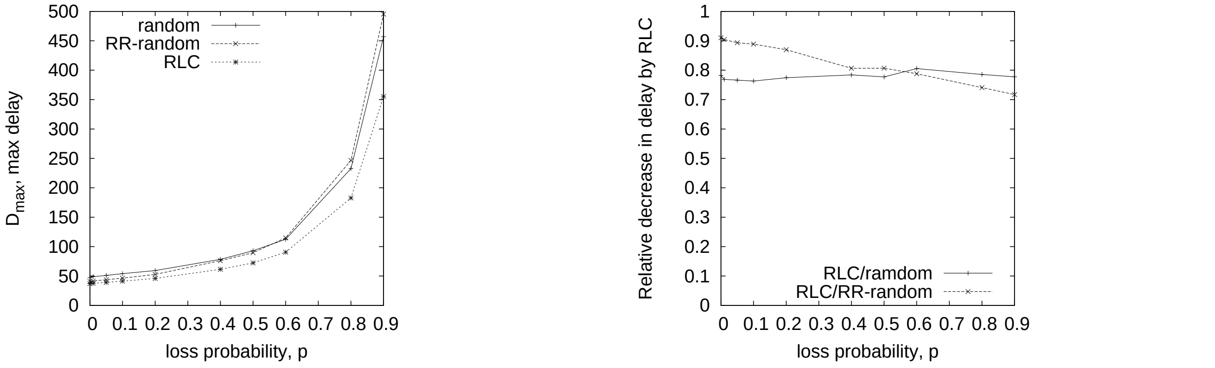
The results show that even with similar transmission numbers, RLC scheme is still able to out perform non-coding forwarding scheme, as the random mixing allows faster and more even propagation of independent information through the network. The results for limited relay buffer case further establish the usefulness of RLC scheme in decreasing block delivery delay when both transmission power and buffer is constrained.

4.4 Effect of Lossy Channel

In this section we consider the benefit of RLC under lossy channels. We assume that each transmission is lost with probability p, and focus on the SS_SD case with a block size of $K = 10$. The results are plotted in Fig.8. We observe that random scheme out-performs RR_random scheme when the loss rate is greater than 0.6. This is because random losses at the source node makes round robin scheduling less effective. We also observe that as the loss rate increases, the benefit of RLC increases. As Figure **??** shows, the effect of lossy channel is equivalent to a channel with smaller effective bandwidth.

5 Multiple Generations Case

In previous section, we have examined the characteristic of RLC scheme, showing it allows faster delivery of a block of packets under bandwidth constraint. We also found that as nodal buffer becomes more and more constrained, RLC is able to maintain almost the same delay performance, at the cost of making more transmissions. Although limiting token numbers leads to larger delay for both RLC and non-coding scheme, RLC scheme is able to achieves better transmission power versus delay tradeoff.



(a) D_{max} under varying loss probability

(b) Relative benefit of RLC under varying loss probability

Figure 8: Effect of lossy channel for SS_SD with $K = 10$ packets

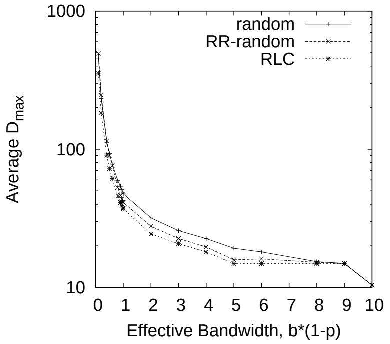


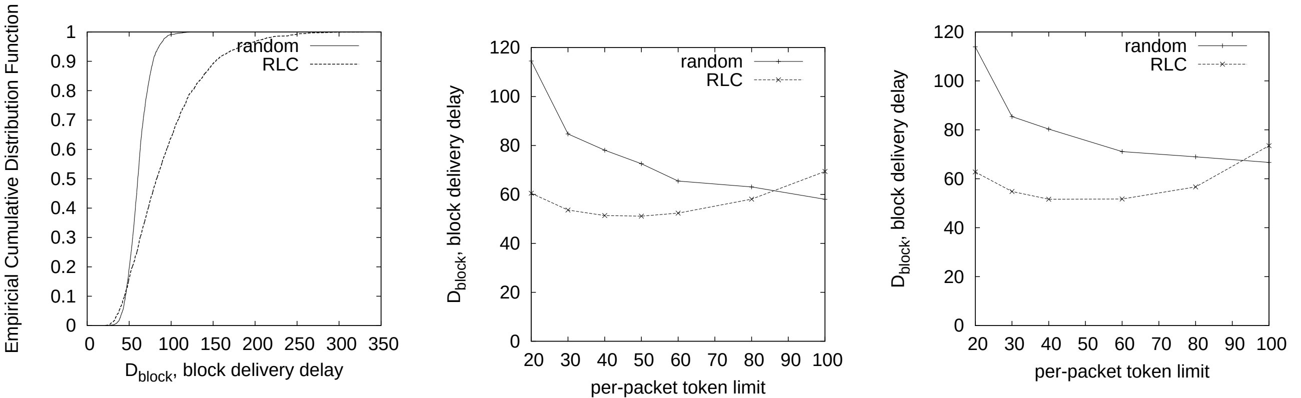
Figure 9: Average block delivery delay under varying bandwidth/loss rate

The natural next question to ask is whether the benefit of RLC remains when one moves from such single generation case to more realistic case where there are multiple continuous flows in the network. We address this question in this section by considering the scenario where there are multiple asynchronous continuous unicast flows in the network. In what follows, we first introduce the traffic process and scheduling schemes, and then present the results for the two scenarios previously considered: when only bandwidth is constrained, and when both bandwidth and buffer are constrained.

5.1 Settings: Traffic Process and Scheduling Schemes

We assume there are N flows in the network, with each node being the source of one flow and the destination of one other flow. Each source node generates independently a block of $K = 10$ packets according to Poisson process with rate λ. Note that the total packet arrival rate to the network is given by $NK\lambda$. We only consider applying RLC to packets belonging to the same block, i.e. each block forms a generation.

As we focus on the understanding of the benefit of RLC, not scheme design, we adopt simple randomized scheduling for both schemes. For non-coding scheme, when a node meets another node, it randomly selects a packet from the set of packets that it carries while the other node does not have, and forwards it. For RLC scheme, the node first randomly chooses a generation from the set of generations that it carries which have some useful information for the other node, and then generates a random linear combination for this



(a) Empirical CDF of D_{block} under $\lambda = 0.00045$

(b) D_{block} under different token limit

(c) D_{block} under different token limit, $B = 5$

Figure 10: Block delivery delay under multiple generation case

generation to forward. For both cases, priorities are given to the packets/generations destined to the other node; furthermore, among such packets/generations, those originated from the node are served first.

5.2 Benefit of Coding under Bandwidth Constraint

We have seen that under bandwidth constraint, for one single generation, RLC achieves lower delay than non-coding scheme, because RLC is able to take advantage of a larger number of contacts. We perform simulation studies under varying block arrival rate with bandwidth constraint $b = 1$. We observe that RLC only has benefit when the traffic rate is low; and performs worse than non-coding scheme when the traffic rate is high, as shown in Fig.10(a) which plots the empirical cumulative distribution function (CDF) of D_{block} under $\lambda = 0.00045$.

The reason is two-fold. First, for non-coding scheme, when the arrival rate λ increases, the number of different packets in the network increases and it is more likely that two nodes have some useful information to exchange when they meet, therefore the gain of RLC is smaller. Secondly, as we have shown in Fig.2.(a), RLC generates more transmissions for each generation. When the block arrival rate is high and there are many generations in the network at the same time, these different generations start competing for the bandwidth. In fact an optimal scheduling should favor a new generation over an old generation, as an old generation has a larger number of combinations spreaded in the network (and with high probability of being already delivered). But currently implemented random scheduling scheme does not consider this optimization.

The tradeoff between number of transmissions and average delay shown in Fig.7 suggests a way to deal with this resource contention problem. The figure shows that RLC can achieve the same delay as non-coding with a significantly lower number of transmissions (left part of the curve), so we expect significant benefit by appropriately limiting copies made for a generation. Fig.10(b) shows that this is the case. Fig.10(b) plots the average D_{block} achieved for RLC and random schemes under block arrival rate of $\lambda = 0.00045$, when the per-packet token limit is varied between 20 and 100. In particular there is an optimal token limit value for RLC scheme, between 40 and 50 token. For higher values, the contention degrades the performance, while for lower values some useful meeting cannot be exploited because all the tokens have been consumed. For non-coding scheme under this arrival rate, the contention is not significant and the reduction of the number of tokens incurs a larger delay. We do observe that under a higher block arrival rate, non-coding scheme also benefits from limiting the number of copies).

How to set the per packet token limit based on bandwidth constraint and block arrvial rate is an open question. We can estimate an upper bound of the number of transmissions that can be made for each packet as the ratio between the total bandwidth available in the networks, $N(N-1)\beta$, and the total arrival rate, $NK\lambda$. For the specific setting considered here, this value is equal to 100.

5.3 Bandwidth and Buffer Constrained Case

We have seen in Section 4.2 that for a single generation case, RLC is especially useful when buffer is constrained, we now consider the scenario when there are multiple generations in the network. As usual, we assume that each node has limited buffer for storing relay packets, but unlimited buffer for storing its own source packets. Since the source node always stores a packet until it is known to be delivered, there is no packet loss. When a node receives a combination and its buffer is full, it first randomly selects one generation from the generations in its buffer that have the highest rank. If the new combination is for the chosen generation, the new combination is combined with an existing combination within that generation. Otherwise, the node compresses the matrix of the selected generation by one, and the new combination is inserted to its generation's matrix.

When both bandwidth and buffer are constrained, limiting number of transmissions made for a generation becomes even more important for RLC scheme. As Fig.6(b) in Section 4.2 shows, under a single generation case, RLC scheme makes much more transmissions than non-coding scheme. Therefore, when there are multiple generations in the network, resource contention is even greater than when buffer is not constrained. We expect that using token scheme allows to allocate bandwidth and buffer space more evenly among different generations. We simulate the case of block arrival rate of $\lambda = 0.00045$, and every node only store $B = 5$ relay packets (combinations) under various token limits. As Fig.10(c) shows, the RLC achieves lower block delivery delay than non coding scheme, reducing the delay by about 22.5%.

6 Summary

We have studied the benefits of applying RLC to unicast application in mobile DTN in this paper. For the case where there is a single generation in the network, we found that RLC applied to a block of data destined to the same destination achieves minimum block delay with high probability. Larger gain is achieved by RLC scheme when furthermore buffer space is constrained. Although RLC scheme makes more transmissions, by using token limit scheme, RLC scheme can achieve better transmission power/delay tradeoff than non-coding approach. When there are multiple generations in the network, under appropriately chosen token limit, RLC scheme achieves slight gain over non-coding scheme under only bandwidth constraint, and a significant gain when nodal buffer is also constrained.

References

[1] S. Acedanski, S. Deb, M. Medard, and R. Koetter. How good is random linear coding based distributed networked storage. In *Netcod*, 2005.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. on Information Theory*, 46:1204–1216.

[3] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. *Allerton Conference on Communication, Control, and Computing*, October.

[4] S. Deb, C. Choute, M. Medard, and R. Koetter. Data harvesting: A random coding approach to rapid dissemination and efficient storage of data. preprint.

[5] S. Deb and M. Medard. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. 2004. Proc. Allerton.

[6] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran. Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes. In *Symposium on Information Processing in Sensor Networks (IPSN '05)*.

[7] Christina Fragouli, Jean-Yves Le Boudec, and Jorg Widmer. Network coding: An instant primer. Technical Report LCA-REPORT-2005-010, 2005.

[8] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. *Infocom*, 2005.

[9] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. In *Performance*, October 2005.

[10] Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. to appear in IEEE/ACM Transactions on Networking.

[11] Sushant Jain, Michael Demmer, Rabin Patra, and Kevin Fall. Using redundancy to cope with failures in a delay tolerant network. In *SIGCOMM*, 2005.

[12] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. In *Journal of Computer and System Sciences, Volume 64,Special issue on STOC 2000*, 2002.

[13] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *ACM workshop on Delay Tolerant Networking*, 2005.

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM workshop on Delay Tolerant Networking*, 2005.

[15] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke Univ., 2000.

[16] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. *ACM workshop on Delay Tolerant Networking*, 2005.

[17] J. Widmer and J.-Y. Le Boudec. Network coding for efficient communication in extreme networks. *WDTN*, 2005.

[18] J. Widmer, C. Fragouli, and J.-Y. Le Boudec. Energy-efficient broadcasting in wireless ad-hoc networks. In *Netcod 2005*, April 2005.

[19] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. Technical Report 2005-44, UMASS Computer Science. ftp://gaia.cs.umass.edu/pub/Zhang05_epidemic_TR.pdf.