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
{ellenz,kurose,towsley@cs.umass.edu}
[§]Università degli Studi di Palermo
giovanni.neglia@tti.unipa.it

***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 the case where there is a single block of K packets propagating in the network and then consider the case where blocks of K packets arrive according to a Poisson 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 needed to deliver the block of data. We find through simulation that RLC achieves smaller block delivery delay than non-network-coded packet forwarding under bandwidth constraint, and the relative benefit 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 improvement 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.**

I. Introduction

Epidemic routing ([14], [12], [13], [6], [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 copy of the packet; newly infected nodes, in turn, behave similarly. The destination receives the packet when it first meets an infected node.

Random Linear Coding (RLC) [8] is a form of network coding [2] where each network node, rather than forwarding packets unchanged along the path from source-to-destination, forwards random linear combinations of the data it has received. First introduced in [8] for multicast, RLC has later been applied to networked scenarios including P2P content distribution [5], practical multicast communications [3], gossip protocol [4], distributed storage [1], and broadcast in mobile or static wireless network (including mobile DTN) [16], [17].

In this paper, we investigate the use of RLC in epidemic routing for unicast applications in mobile DTNs. In this case there are different possible ways to combine packets. We first consider three possibilities in the simple case of 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. 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, and a smaller average delay than non-coding scheme. We find through simulation that this benefit increases further when buffer space within the DTN nodes is limited. We also demonstrate the “price” to be paid for the decreased delay is a larger number of epidemically-spread copies of data in the DTN. Using a token scheme to limit the total number of transmissions made, we find that RLC achieves a smaller block delivery delay than non_coding scheme under similar number of transmissions.

We then consider the case where blocks of K packets arrive to the source nodes according to a Poisson bulk arrival process. We find that at a relatively high load, RLC without token limit leads to higher resource contention and larger delays. When a token scheme is used to control the propagations of each block, RLC offers a slight improvement over the non-coded forwarding when only bandwidth is constrained, but more significant benefits when both bandwidth and buffers are constrained.

We note that our result that RLC scheme decreases block delivery delay for the single block case is similar in spirit to [4], where the authors showed that applying RLC to gossip protocol achieves order-optimal delay to spread multiple messages to the whole network. Our study differs from [4] (and [16], [17]) in that we study unicast applications, for which there exists a trade-off between delay and number of transmissions. Our study of token limited scheme reveals that RLC achieves smaller block delivery delay than non-coding scheme under similar number of transmissions, and thus remains beneficial under multiple generation case. For unicast applications in wireless network, [10] showed that network coding scheme can increase the throughput of unicast by leveraging the broadcast nature of the wireless medium. For the sparse mobile network that we study in this paper, the probability of having multiple nodes in a node's transmission range is very small.

Several previous research efforts have applied *source-based* (i.e., non-network-coded) erasure codes to DTNs. [15] proposed 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, [9] considered how to allocate the source-erasure-coded blocks to these paths.

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

II. 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 around 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 determine if they have useful information to exchange and, if so exchange it. We detail this process with reference to the two mechanisms we are going to compare: traditional non-coded forwarding and RLC-forwarding.

Non-coded forwarding: When two nodes meet, each of them randomly selects one or more packets, depending on the bandwidth, 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. Typically, F_{2^8} (i.e.,F_{256}) is used [16]. 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 K x r matrix made up of the coefficients of the r combinations as the node's *encoding matrix*.

Under RLC, when two nodes meet, they first transfer their encoding matrices to each other. Each node, based on the matrix, checks if it has useful information for the other node[1]. 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: $f_{new} = \sum_{j=1}^{r} \beta_j f_j$. This new combination, f_{new}, along with its coefficients in terms of the original packets, is forwarded to the other node[2]. When a node (e.g., the destination) reaches rank K, it can decode the original K packets through matrix inversion. Notice that RLC incurs storage overhead for storing coefficients for each combinations, and it also requires more computation to check if one node has useful information for the other and to decode the combinations.

Under epidemic routing, when a packet is delivered to the destination, a recovery scheme can be used to delete copies of the packet from the network [7]. We consider the 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, when a generation is delivered to the destination, an antipacket for the generation is generated and propagated to delete buffered combinations of the generation. We assume that the recovery scheme is not subject to bandwidth or 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. In particular, we define *block delivery delay* (D_{block}) as the time from the arrival of the block to the source, to the delivery of the entire block to the destination. 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 average number of packet copies or combinations being transmitted within the network, as this is a measure of resources consumed (bandwidth, transmission power, buffer) within the DTN.

III. Simulation Setting

In this section, we describe our simulation setting. We model the the movement of nodes and their meeting as follows: the inter-meeting processes for all the node pairs follow independent and identical Poisson processes. [6] has shown that under the random waypoint and random direction models, the inter-meeting time between a pair of nodes follows a Poisson

[1] In fact, if a node has at least one combination that cannot be linearly expressed by the combinations stored in another node, it has useful information for the other node.

[2] 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 [4].

process when node velocity is relatively high compared to the region size, and the transmission range is relatively small. We have also performed simulations where mobile nodes move according to random waypoint model 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 $q = 701$.

IV. 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 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.

A. 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_{block} with high probability.

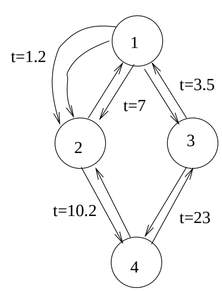


Fig. 1. Random graph representing the contacts between nodes, $b = 1$

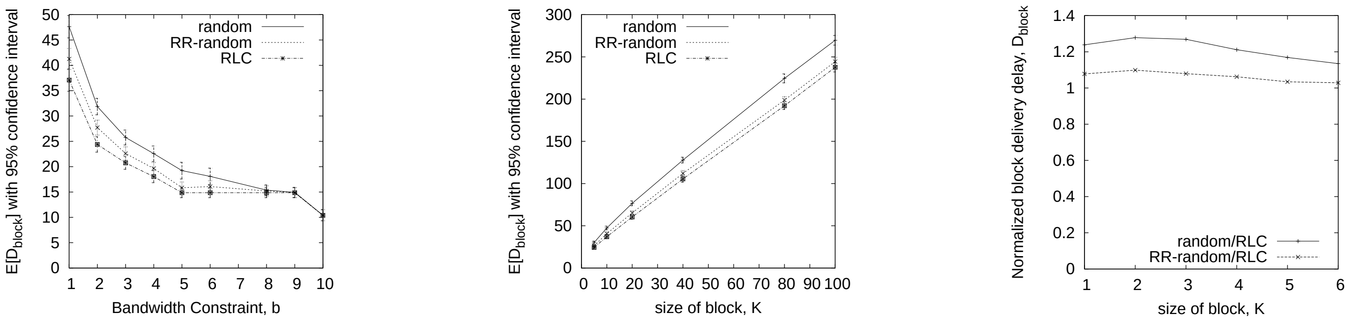
We provide an intuitive argument using a random multi-graph constructed as follows (Fig.1): there are N vertices, each corresponding to one mobile node, and for each contact between a pair of nodes that can exchange b packets in each direction, add b directed edges in each direction between the corresponding vertices. Edges are labeled with the time that the contact occurs. A *time-respecting path* in such a network is a path in the graph where the successive edges have increasing timestamps. A set of paths are edge-disjoint if they do not share edges. Previous work [11] has studied connectivity and inference problems for such *temporal networks* that arise from applications such as communication in distributed networks and epidemiology etc.. Our model further captures the finite capacity of the contact.

For the SS_SD case, where the source node initially has K packets to send, the time to deliver these K packets cannot be smaller than the time when there are K edge-disjoint 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 edge-disjoint 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" encoded-packets. As the number of independent coded packets is much larger than K, the probability that nodes forward a coded-packet that is useless to the destination is much smaller than in non-coded scheme.

We use the 4-node network in Fig.1 to illustrate this idea. Here on each contact, one packet can be transmitted in each direction. Assume that node 1 generates two packets m_1, m_2 destined to node 4 at $t = 0$. Without applying network coding, node 1 forwards m_1, m_2 to node 2, and one of the packets (say m_1) to node 3. When nodes 2 and 4 meet at $t = 10.2$, node 2 randomly selects a packet and delivers to node 4 (given that the node has no global knowledge of past and future contacts for other nodes). With probability 0.5, packet m_2 is forwarded to node 4. As a result, when node 3 meets node 4 at $t = 23$, it has no useful information for node 4. On the other hand, if RLC is used, suppose source node 1 forwards random linear combination c_1, c_2 to node 2, and c_3 to node 3. On meeting node 4, node 2 generates a random linear combination c_{12} of c_1, c_2 and forwards it. As long as c_{12} and c_3 are independent (with probability $1 - 1/q$), node 4 can decode the two original packets after node 3 delivers c_3 at time $t = 23$.

As a quantitative analysis of delivery delay is difficult due to the randomness 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 first highlight several characteristics of RLC compared to a non-coded approach. Fig.3 depicts 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 for SS_SD under $N = 101, K = 10$ in a particular run. We observe that RLC incurs more transmissions and storage occupancy in the network. There are two factors causing this: first, RLC allows faster propagation of information in the network, as two nodes more frequently have useful information to exchange when they meet; second, under RLC 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).



(a) $E[D_{block}]$ under varying bandwidth (b) $E[D_{block}]$ under different block size (c) Benefit of RLC under different block size

Fig. 2. RLC benefit under SS_SD

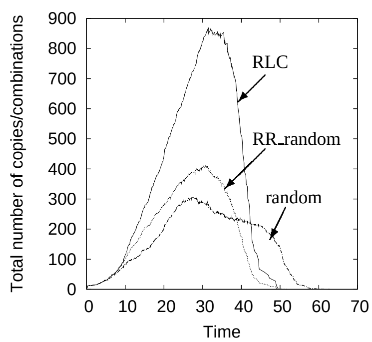


Fig. 3. RLC scheme achieves faster propagation

We next explore the benefit of RLC relative to the non-coded case under varying bandwidth constraints. Fig.2(a) plots the $E[D_{block}]$ for SS_SD with $K = 10$ under varying bandwidth constraints. We note that the $E[D_{block}]$ reported throughout Section IV is the average from 50 different simulation runs. The figure shows that RLC achieves a smaller $E[D_{block}]$ than both random and RR_random schemes. Furthermore, the relative benefit of RLC increases as bandwidth decreases.

Fig.2(b) shows the dependence of $E[D_{block}]$ on the block size for the SS_SD case under 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). Fig.2(c) plots the average block delivery delay under non_coding schemes normalized with respect to that of RLC under different block sizes. We observe that as the block size increases, the relative benefit of RLC over non-coding scheme decreases.

Due to space constraint, we do not show results for MS_SD and MS_MD case here [18]. We note that the benefit achieved by RLC for the MS_SD case is smaller than for the SS_SD case. 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, we find that 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.

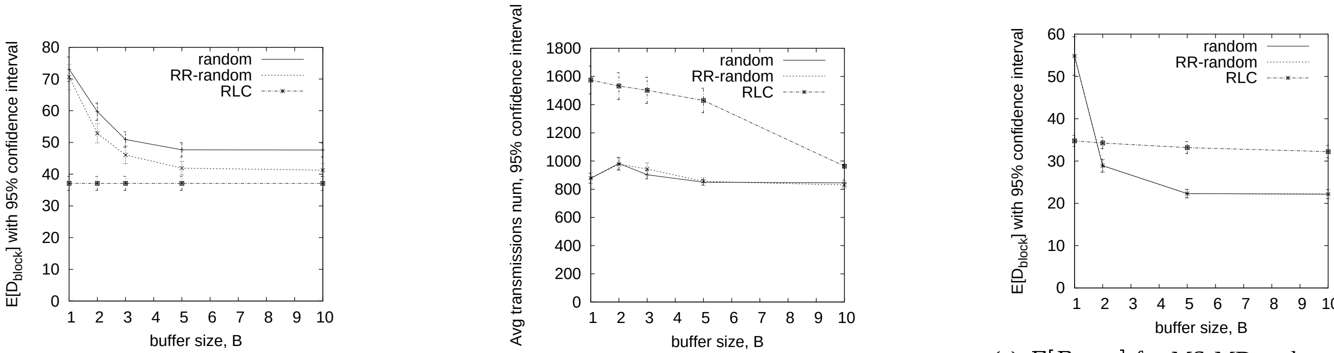
B. Benefit of Coding under Bandwidth and Buffer Constraints

Thus far, we have assumed that nodes have unlimited buffer capacity. We now consider the case where 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.4(a) shows that, for the SS_SD case (with $K = 10$), as buffer sizes decrease, the performance of RLC deteriorates only slightly; while the performance of the non-coding schemes degrade quickly. To explain the benefit of RLC, we examined the simulation trace closer. We found that in the absence of coding, while some packets spread quickly to a large number of nodes, other packets propagate much slower. This can be explained by the adopted random selection scheme: the more copies a packet has in the network, the more likely the packet is forwarded to other nodes and cause a copy of some other packet to be dropped. Under RLC, different packets are mixed randomly by nodes, therefore, when a node's buffer is full, equal amount of information is dropped for each packet. As a result, compared to the non-coding approach, RLC allows independent information of the K packets in the block to propagate more evenly, leading to a much smaller block delivery delay.

Again, we note that the improvement in delay performance of RLC is achieved at the cost of more transmissions made as shown in Fig.4(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 where a node can receive different combinations of a generation without increasing rank over the buffer size B.

For MS_SD, we observe similar performance gains of RLC (not shown here [18]). Under MS_MD ($K = 10$), where coding is applied to packets sent by different sources to different destinations, when the buffer is very constrained



(a) $E[D_{block}]$ for SS_SD under varying buffer

(b) Number of transmissions made for SS_SD

(c) $E[D_{block}]$ for MS_MD under varying buffer size

Fig. 4. Under bandwidth and buffer Constraints

($b = 1$ for this setting), RLC out-performs the non-coding random scheme (Fig.4(c)).

C. Controlling Transmission Power Consumption of RLC

We have seen that RLC decreases the delay to deliver a block of data, or collect multiple packets from different sources, at the "cost" of consuming more buffer space and transmission power. Can RLC achieve smaller block delivery delay than non-coded scheme under the same transmission power consumption ? We address this question in this section.

To limit the number of copies made for a packet, we adopt a token-based scheme, extending the *spray and wait* scheme proposed in [13], [12]. Under this scheme, the source nodes assign every new packet a certain number of tokens (which we refer to as *per-packet token number*). When the packet is copied to another node, half of the tokens are assigned to the new copy. When a packet copy has only a single token remaining, it can only be forwarded to the destination. The total number of copies of a packet is thus bounded by the initial token number. 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. We also extend the notion of tokens to RLC by associating a token number with each generation, which equals the product of the number of packets in the generation and the per-packet token number. Instead of splitting tokens in equal halves when making a new copy and when two nodes meet, the token number for a generation are allocated to two nodes in proportion to their ranks (i.e., the amount of information the nodes carry about the generation).

We run simulations for SS_SD case with $K = 10$ with different per-packet token numbers. We find that RLC achieves better transmission/delay tradeoff than non_coding, as shown in Fig.5. The figure plots for SS_SD case with $K = 10$, the number of transmissions versus delay tradeoff achieved when the per-packet token limits are varied between 4 and infinity. It shows that even with similar transmission numbers, RLC still achieves a smaller average block delivery delay than a non-coding forwarding scheme, as the random mixing allows a faster and more even propagation of independent information through the network. Similar results for bandwidth and buffer constraints case further establish the benefits of RLC scheme in decreasing block delivery delay, when both transmission power and buffer is constrained [18].

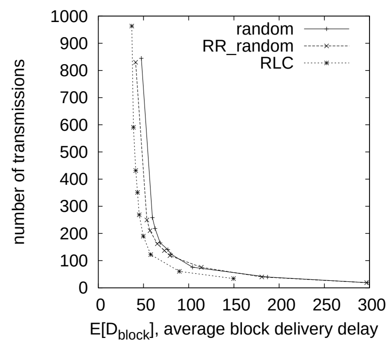


Fig. 5. Transmission power vs delay trade-off achieved with different token number

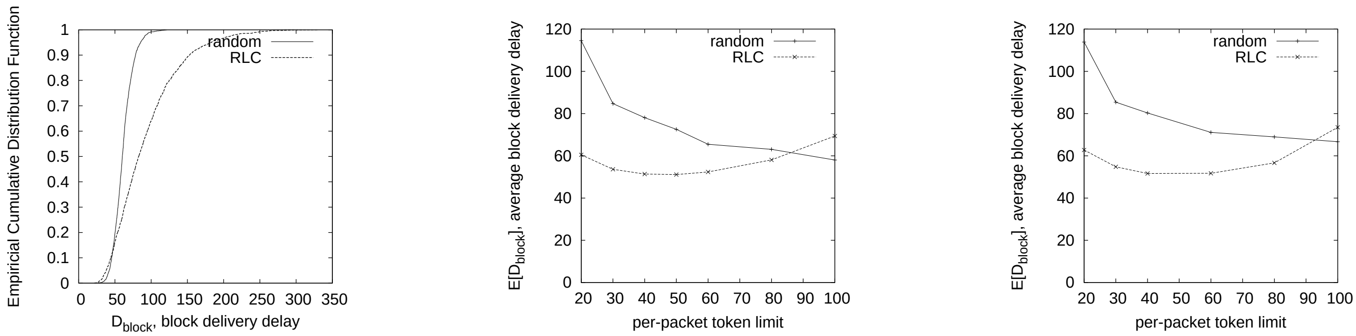
V. Multiple Generation Case

In this section, we investigate whether RLC is beneficial in a more complex scenario where many generations are present at the same time in the network. In what follows, we first introduce the traffic process and scheduling policies, and then present the results for the two scenarios previously considered: when only bandwidth is constrained, and when both bandwidth and buffer are constrained.

A. 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 blocks of $K = 10$ packets according to Poisson process with rate λ. We only consider applying RLC to packets belonging to the same block, i.e. each block forms a generation.

As our focus is not on designing an optimal scheduler, but rather understanding the benefit of RLC, we adopt simple randomized scheduling. In the absence of coding, 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. In RLC, the node first randomly



(a) Empirical CDF of D_{block} under $\lambda = 0.00045$ (b) $E[D_{block}]$ under different token limit (c) $E[D_{block}]$ under different token limit, $B = 5$

Fig. 6. Block delivery delay under multiple generation case

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 generation to forward. In 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.

B. Benefit of Coding under Bandwidth Constraint

We have seen that under bandwidth constraints, for one single generation, RLC achieves lower delays than non-coding. We perform simulation studies for different block arrival rates with bandwidth constraint $b = 1$. We observe that RLC only benefits when the traffic rate is low; and performs worse than non-coding scheme when the traffic rate is high, as shown in Fig.6(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, 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. Second, as 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 scheduler should favor a new generation over an old generation that has a larger number of combinations spread over the network (and with high probability of being already delivered). But the scheduler we considered do not take into account this potential improvement.

The transmission-delay tradeoff shown in Fig.5 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.6(b) shows this to be the case. Fig.6(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 meetings 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 (note that we 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 arrival 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.

C. Bandwidth and Buffer Constrained Case

We now consider the case when both bandwidth and buffer are constrained. 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 selects randomly one generation from the generations in its buffer that have the highest rank. If the new combination is for the chosen generation, it is combined with one of the other combinations for the generation. Otherwise, one of the combination of the selected generation is replaced by the new combination and the generation's matrix is updated.

We observe that under bandwidth and buffer constraints, limiting the number of transmissions made for a generation becomes even more important for RLC scheme. We have seen that under a single generation case, RLC scheme makes much more transmissions than non-coding scheme (Fig.4(b)). Therefore, when there are multiple generations in the network, resource contention is even greater than when the buffer is not constrained. We expect that using a token scheme allows one 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.6(c) shows, RLC achieves lower block delivery delay than non coding, reducing the average block delivery delay by about 22.5%.

VI. 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 also 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.

As to future work, we plan to consider the overhead of RLC scheme in comparison to the benefit achieved, and study the case where packets arrive according to Poisson process (other than the bulk Poisson process considered in the paper). We will also investigate the problem of whether network coding can increase network throughput in DTN.

Acknowledgement

This research was supported in part by the National Science Foundation under award number ANI-0085848, EIA-0080119, the Engineering Research Centers Program, award number EEC-0213747001, and Italian MIUR project Famous. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] S. Acedanski, S. Deb, M. Médard, 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 and M. Médard. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. 2004. Proc. Allerton.

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

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

[7] 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.

[8] T. Ho, R. Hoetter, M. Médard, D. R. Karger, and M. Effros. The benefits of coidng over routing in a randomized setting. In *IEEE International Symposium on Information Theory*, 2003.

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

[10] S. Katti, D. Katabi, W. Hu, R. Hariharan, and M. Médard. The importance of being opportunistic: Practical network coding for wireless environments. In *Allerton Conference on Communication, Control, and Computing*, Sep 2005.

[11] 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.

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

[13] 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.

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

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

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

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

[18] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. Technical Report 2006-15, UMASS Computer Science. ftp://gaia.cs.umass.edu/pub/Zhang05_netcoding_TR.pdf.

[19] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *IFIP Networking*, 2006.