Benefits of Network Coding for Unicast Application in Disruption Tolerant Networks

Xiaolan Zhang, *Member IEEE*, Giovanni Neglia, *Member IEEE*, Jim Kurose, *Fellow IEEE*, Don Towsley, *Fellow IEEE*, Haixiang Wang

***Abstract*—In this paper, we investigate the benefits of applying a form of network coding known as Random Linear Coding (RLC) to unicast application in Disruption Tolerant Networks (DTNs). Under RLC, nodes store and forward random linear combinations of packets as they encounter each other. We first consider RLC applied to a single group of packets originating from the same source and destined for the same destination. We develop an algorithm for calculating the minimum group delivery delay (i.e., time to deliver the last packet in the group), and prove a lower bound on the probability that the RLC scheme achieves the minimum delivery delay. Compared to the non-coding scheme, the RLC scheme achieves a smaller average group delivery delay due to its increased randomness, but fares worse in terms of average delivery delay and incurs more network transmissions. However, when replication control is employed, RLC schemes reduces the average group delivery delay without increasing the number of transmissions. We also investigate the impact of resource constraints, control signaling, and real mobility traces on the benefit of the RLC scheme. Finally, we show that coding together packets with different destinations is in general less beneficial. With multiple continuous flows in the network, the RLC scheme, even when applied only to packets from the same flow, needs to be employed with a carefully tuned replication control token limit in order to achieve improvements in average delay. More significant RLC benefit is observed when buffer space is limited.**

I. Introduction

In recent years, wireless communication technologies have been increasingly deployed in environments where there is no communication infrastructure, as evidenced by the many efforts to build and deploy wireless sensor networks for wildlife tracking [23], [15], underwater sensor networks [35], [36], disaster relief team networks, networks for remote areas or for rural areas in developing countries [1], [12], [2], vehicular networks [7], [20] and Pocket-Switched Networks [19]. Without infrastructure support, such networks solely rely on peer-to-peer connectivity between wireless radios to support data communication. Due to limited transmission power, fast node mobility, sparse node density and frequent equipment failures, many such networks exhibit only intermittent connectivity. *Disruption Tolerant Network* (DTN, or *Delay Tolerant Network*), refers to such a network where there is often no contemporaneous path from the source node to the destination node. End-to-end communication in DTNs adopts a so-called "store-carry-forward" paradigm [43]: a node receiving a packet buffers and carries the packet as it moves, passing the packet on to new nodes that it encounters. The packet is delivered to the destination when the destination meets a node carrying the packet. In addition to intermittent connectivity and dynamic topologies, DTNs often face additional challenges due to severe resource constraints. For small mobile nodes carried by animals or human beings, buffer space, transmission bandwidth and power are very limited; for mobile nodes in vehicle based networks, buffer space or power constraints are generally not constrained, but transmission bandwidth is still a scarce resource. To address the unique challenges of DTNs, a plethora of routing schemes have been proposed for DTNs ([43], [42], [39], [15], [40], [41], [5]): some of these papers explore the trade-off between routing performance and resource consumption, whereas others attempt to optimize routing performance under some given resource constraints.

The work by Ahlswede *et al.* [4] demonstrated the benefit of coding at intermediate nodes in terms of approaching the admissible coding rate region for multicast applications, and initiated a new field in information theory, i.e., network coding. Among the many works that followed, a substantial amount of research has studied the benefits of network coding for multicast, broadcast and unicast applications in wireless networks. Although a DTN is a special type of wireless network, due to its distinct characteristics, some benefits of network coding for general wireless networks do not hold. First, due to the dynamically changing topology of a DTN, the results obtained in [34], [47] for multicast application in static wireless networks are not directly applicable. Second, DTNs are sparse with each node usually having at most one neighboring node at any instance of time, therefore the benefit of network coding in increasing network throughput (by leveraging the broadcast nature of wireless transmission) is negligible in DTNs.

On the other hand, there are new opportunities for network coding in DTNs. The rapidly changing topology and the lack of infrastructure require DTN routing schemes to be *distributed*; and the limited connectivity and bandwidth also require DTN routing schemes to be *localized*, i.e., using only limited knowledge about the local neighborhood. Network coding has been shown to facilitate the design of efficient distributed schemes not only for routing in wireless networks but also for P2P content diffusion [13].

Existing research on the application of network coding to

Xiaolan Zhang is with the Department of Computer and Information Science, Fordham University. Haixiang Wang was with the Department of Computer and Information Science, Fordham University when this work was done, Bronx NY 10458 USA e-mail: {xzhang,ahawang}@fordham.edu.

Giovanni Neglia is with INRIA Sophia Antipolis, France, email: giovanni.neglia@sophia.inria.fr

Jim Kurose and Don Towsley are with the Department of Computer Science, University of Massachusetts, Amherst, MA 01008, USA email: {kurose,towsley}@cs.umass.edu

DTNs has focused on applying Random Linear Coding (RLC), a special form of network coding [17], to broadcast and unicast communication. In this paper, we use the term *RLC scheme* to denote a DTN routing scheme that employs RLC, and use the term *non-coding scheme* to denote a traditional routing scheme. For broadcast application, Widmer *et al.* ([45], [46]) showed that the RLC scheme achieves higher packet delivery rates than the non-coding scheme with the same forwarding overhead. For unicast applications, our earlier work [48] showed through simulation that RLC schemes achieve faster delivery of a group of packets than non-coding schemes at the expense of large number of transmissions in the network, and when the replication control is employed, RLC schemes improve the trade-off between delivery delay and resource consumption in terms of the number of transmissions made, and remains useful when there are multiple unicast flows with group of packets arriving according to a Poisson process. Lin *et al.* proposed and analyzed a different replication control scheme ([30]), and proposed Ordinary Differential Equation (ODE) models for estimating delivery delay and number of transmissions for RLC schemes and non-coding schemes ([31]), both for a single group of K packets.

This paper presents new contributions that improve our understanding of the benefits of network coding in DTNs unicast application both theoretically and practically. Leveraging the event-driven graph model for DTNs ([16]), and existing results on static graphs ([18], [26]), we propose an algorithm to calculate the minimum time to deliver a group of packets (for SS_SD and MS_SD cases), and prove a lower bound on the probability that RLC schemes achieve the minimum delivery time. Interestingly, the proof of the lower bound also demonstrates the connection between the RLC benefits in resource constrained DTNs and traditional network. Furthermore, we discuss the design space of DTN unicast routing schemes, and consider the impact of resource constraints, control signaling, real mobility trace and different generation management (both intra-flow and inter-flow coding) on the benefit of RLC.

The remainder of this paper is structured as follows. In Section II, we introduce the network model, and performance metrics considered in this paper, overview the non-coding schemes and the basic operation of RLC schemes, and discuss the design space of DTN unicast routing schemes. Section III studies the benefit of the RLC scheme over the non-coding scheme for a single source case, i.e., a group of packets originated from a single source and destined for a single destination. Section IV extends the study to multiple source case, and investigates the alternative generation management and the case of multiple continuous unicast flows. Section V reviews related work. Finally, Section VI concludes this paper.

II. Background

In this section, we first present the network model, traffic setting and performance metrics studied in this paper. We then describe the general approach to unicast routing in DTN with and without Random Linear Coding. Last, we provide a discussion about the design space for DTN routing schemes.

notation	meaning	simulation setting
N	number of nodes in the network	101
$\mathcal{V}$	the set of nodes	
$\mathcal{L}$	DTN contact trace	
β	pair-wise contact rate	0.0049
K	generation size	10
λ	group arrival rate to each flow	varies
b	#. of packets can be exchanged in each direction during a contact	1
B	#. of relay packets a node can store	varies
$\mathbb{F}_q$	finite field, $q = p^n$, p is a prime, n is a positive integer.	$q = 2^8$
D_g	time to deliver a group of packets	N/A
C	per-packet token number	varies
C_g	per-generation token number	varies

TABLE I
Tables of Notations

Table I summarizes the notation used in this paper and the default settings used in the simulation studies.

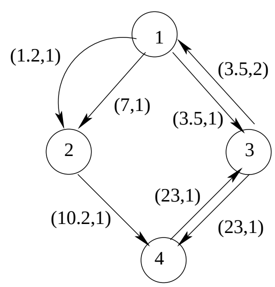
A. Network Model

Consider a network consisting of a set of N mobile nodes, denoted as $\mathcal{V}$, moving independently in a closed area. Each node is equipped with a wireless radio with a common transmission range so that when two nodes come within transmission range of each other (i.e., they *meet*), they can exchange packets. The *contact duration* is the time duration of this transmission opportunity, while the *inter-contact time* is the duration of the time interval between two consecutive meetings, i.e. measured from the time that the two nodes go out of the transmission range of each other until the next time they meet again. We refer to the list of node-to-node contacts, sorted in temporal order, within a DTN during a certain time interval as ***a DTN contact trace***, denoted as $\mathcal{L} = l_1, l_2, l_3, ...$, where each node-to-node contact, l_i, is a tuple $(t(l_i), s(l_i), r(l_i), b(l_i))$ where $t(l_i)$ denotes the time of the contact, $s(l_i)$ and $r(l_i)$ denote respectively the sending and the receiving node of the contact, and $b(l_i)$ denotes the number of packets that can be transmitted during the contact[1].

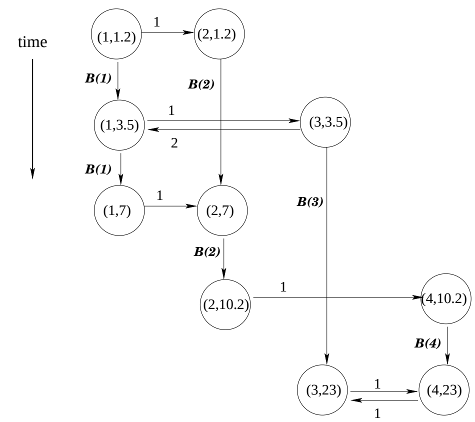
As for the buffer constraint, we assume each node can store an unlimited number of packets originated by itself or destined for itself, but can only carry a limited number of packets for other nodes. We represent the buffer constraint as a function, $\mathcal{B} : \mathcal{V} \to \mathbb{N}$ where $B(u)$ is the number of relay packets that node u can carry.

A contact trace can be represented as a *temporal network* as originally proposed by Kempe *et al.* [25]. The temporal network for contact trace $\mathcal{L}$ is a multi-graph $\mathcal{T}(\mathcal{L}) = \langle \mathcal{V}, \mathcal{E} \rangle$ in which $\mathcal{V}$ denotes the set of nodes in the network, and $\mathcal{E}$ denotes the set of *directed* edges. Each contact $l \in \mathcal{L}$ is represented as an edge, labeled with a pair, $(t(l), b(l))$, i.e.,

[1] In general, contacts can be *directed*, if two independent wireless channels are used for transmissions in the two directions, or *undirected*, if the same wireless channel is used for transmissions in both directions and the total capacity can be arbitrarily divided between them. We focus on the first case in this paper.



(a) Temporal network representation, $\mathcal{T}(\mathcal{L})$



(b) Event-driven graph representation, $\mathcal{G}(\mathcal{L}, \mathcal{B})$

Fig. 1. Graph representations of a DTN contact trace.

the time of the contact, and the number of packets that can be exchanged using the contact. For example, Fig. 1(a) illustrates the temporal network model for a contact trace of a DTN with four nodes during the time interval $[0, 24]$.

Another useful graph representation for a DTN contact trace is the *event-driven graph* first proposed in [16]. As an example, Fig. 1(b) shows the event-driven graph corresponding to the contact trace in Fig. 1(a). The event-driven graph $\mathcal{G}(\mathcal{L}, \mathcal{B})$ for a contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$ is constructed as follows: For each contact $l = (t, u, v, b) \in \mathcal{L}$, two nodes (u, t) and (v, t) are added into graph $\mathcal{G}$, respectively denoting the sending and receiving event of the contact. A directed *inter-node edge* (depicted as horizontal lines in Fig. 1(b)), labeled with b, connects node (u, t) to node (v, t), denoting that up to b packets can be transmitted from node u to v at time t. If two consecutive contacts involving node u occur at t_1 and $t_2 (> t_1)$, a directed *intra-node edge* connecting nodes (u, t_1) to (u, t_2) is added to graph $\mathcal{G}$ (depicted as a vertical line in the figure), with a capacity equal to $\mathcal{B}(u)$, i.e., the maximum number of relay packets node u can store.

The event-driven graph is a static, i.e., time-independent, graph that not only captures the temporal constraint of the contacts, but also represents the bandwidth and buffer constraints. [16] showed that many problems on DTN routing can be solved by applying classic Graph Theory algorithms on this static graph. We use the following proposition (a restatement of Theorem 4 in [16]) in this paper:

Proposition 2.1: There is a feasible routing schedule for delivering K packets originated from u before t_1 to node v by time $t_2 (t_2 \geq t_1)$ under contact trace $\mathcal{L}$ and buffer constraint $\mathcal{B}(\cdot)$ if and only if there is a flow of value K from node (u, t_1) to node (v, t_2) in the event-driven graph $\mathcal{G}(\mathcal{L}, \mathcal{B})$.

To see this, we note that the value of a flow on an inter-node edge (e.g., $(1, 1.2) \rightarrow (2, 1.2)$) equals the number of packets sent during the corresponding contact (e.g., $(1.2, 1, 2, 1)$), whereas the value of a flow on an intra-node edge (e.g., $(2, 1.2) \rightarrow (2, 7)$) corresponds to the number of packets being carried by the node (node 2) during the corresponding time interval ($[1.2, 7]$).

In our simulation studies, we assume homogeneous resource constraint. In particular $\mathcal{B}(u) = B$, for all $u \in \mathcal{V}$, and when two nodes encounter each other, b packets can be exchanged in each direction. Most of our simulation results are obtained under the assumption that pair-wise meeting is described by independent and Poisson processes with rate β. This simplification speeds up the simulations, and is a good approximation on timescales beyond the average time a node spends to cross the region, when nodes move according to common random mobility models (like random waypoint and random direction) and the network is sparse. This observation was first made by [14]. Later works ([24], [8]) have formally proven that the tail of the Complementary Cumulative Distribution Function (CCDF) of the inter-contact time is actually exponentially bounded for many common random mobility models in a finite region. The characteristic time beyond which the inter-contact time exhibits an exponential behavior has been investigated in [8], [9]. Because of its tractability, the Poisson meeting process has been widely adopted specially for modeling purposes ([14], [30]).

B. Traffic Setting and Performance Metrics

We focus on unicast applications where each packet (generated by its source node) is destined to a single destination node.

We assume that each message generated by the application is segmented into a group of packets in order to take advantage of the short contact [37]. We denote the group of packets belonging to a message as $P_i, i = 1, 2, ..., K$, and the delivery delay of packet P_i as D_i for $i = 1, 2, ..., K$. The *group delivery delay*, D_g, is the time from the generation of the message, i.e., the group of packets, to the delivery of the entire group to the destination, and we have $D_g = \max_{1 \leq i \leq K} D_i$. Depending on the specific application, other metrics such as the *average packet delivery delay* (if the application can process each packet individually upon its delivery) and the *average in-order packet delivery delay* (if packets must be processed by the destination in order) for all packets in the group might be more meaningful. The *in-order packet delivery delay* for packet P_i is $D'_i = \max_{1 \leq j \leq i} D_i$, for $i = 1, 2, ..., K$,

For applications that generate small messages, segmenting the message into even smaller packets would lead to a large relative overhead (for packet headers and encoding vectors). In such applications, RLC can be applied to a group of packets whose generation times are close to each other.

As a measure of resources consumed (bandwidth, transmission power, buffering) in the network, we consider the total number of transmissions made within the network for the group. There exists an inherent trade-off between the delivery delay and the number of transmissions made [49], which is be further studied in Section III-C.

C. Non-Coding Routing Schemes

Non-coding based unicast routing schemes for DTNs can be classified as single-copy or multi-copy schemes.

Under a single-copy scheme [41], each packet is *forwarded* along a single path, and at any point in time, there is a single copy of the packet in the network. Single-copy schemes place minimal demand on the node buffer space, and usually incur a low transmission overhead. But when future contact processes are not known in advance, forwarding decisions can later turn out to be wrong and in general lead to suboptimal performance. In such cases, it is often beneficial to use multi-copy schemes to reduce delivery delay and increase the delivery probability at the expense of larger transmission overhead and buffer occupancy.

Under a multi-copy scheme, a packet is *copied* to other nodes to be simultaneously forwarded along multiple paths to the destination, leading to multiple copies of a packet in the network at a given point in time. For example, epidemic routing proposed by Vahdat and Becker [43] floods the whole network in order to deliver a packet. By making use of all transmission opportunities, epidemic routing achieves minimum delivery delay when the network is lightly loaded, but it causes severe resource contention under heavier traffic. Many variations of epidemic routing that trade-off delivery delay for resource consumptions have been subsequently proposed and studied, including K-hop, probabilistic forwarding [15] and spray-and-wait [42], [39], [40].

D. RLC based Routing Schemes

In this section, we describe the basic operation of Random Linear Coding (RLC) based DTN routing schemes.

We assume that all packets have the same payload size equal to S bits. When RLC is used in packet data networks, the payload of each packet can be viewed as a vector of $d = \lceil S/\log_2(q) \rceil$ symbols from a finite field [28], $\mathbb{F}_q$ of size q.

A collection of packets that may be linearly coded together by network nodes is called a *generation*. For example, the K packets that make up an application message can constitute a generation. We denote by $\mathbf{m}_i \in \mathbb{F}_q^d$, the symbol vector corresponding to packet $P_i, i = 1, 2, ...,$. A linear combination of the K packets is:

$$\mathbf{x} = \sum_{i=1}^{K} \alpha_i \mathbf{m}_i, \;\; \alpha_i \in \mathbb{F}_q,$$

where addition and multiplication are over $\mathbb{F}_q$. The vector of coefficients, $\alpha = (\alpha_1, ..., \alpha_K)$ is called the *encoding vector*, and the resulting linear combination, $\mathbf{x}$, is called an *encoded packet*. We say that two or more encoded packets are linearly independent if their encoding vectors are linearly independent. Each original packet, $\mathbf{m}_i, i = 1, 2, ...K$, can be viewed as a special combination with coefficients $\alpha_i = 1$, and $\alpha_j = 0, \forall j \neq i$.

Under RLC schemes, network nodes store and forward encoded packets, together with their encoding vectors. For a generation of size K, the coefficients take up K symbols; while each data packet (original or encoded) takes up $d = \lceil S/\log_2(q) \rceil$ symbols. The relative overhead, i.e., the ratio of the size of the encoding coefficients and the data packet, is $K/(\lceil S/\log_2(q) \rceil) \approx K \log_2(q)/S$.

If the set of encoded packets carried by a node contains at most r linearly independent encoded packets $\mathbf{x}_1, ..., \mathbf{x}_r$, we say that the rank of the node is r. We refer to the $r \times K$ matrix (denoted as $\mathbf{A}$) formed by the encoding vectors of $\mathbf{x}_1, ..., \mathbf{x}_r$ as the node's *encoding matrix*. Essentially, the node stores r independent linear equations with the K original packets as the unknown variables, i.e., $\mathbf{AM} = \mathbf{X}$, where $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_K)^{\mathrm{T}}$ is a $K \times d$ matrix of the K original packets, and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_r)^{\mathrm{T}}$ is an $r \times d$ matrix of the r encoded packets. When a node (e.g., the destination) reaches rank K (i.e., *full rank*), it can decode the original K packets through matrix inversion, solving $\mathbf{AM} = \mathbf{X}$ for $\mathbf{M} = \mathbf{A}^{-1}\mathbf{X}$ using standard Gaussian elimination algorithm[2].

We illustrate data forwarding under RLC schemes using the transmission from node u to node v as an example. Node u generates a random linear combination ($\mathbf{x}_{new}$) of the combinations stored in its buffer $\mathbf{x}_1, ..., \mathbf{x}_r$: $\mathbf{x}_{new} = \sum_{j=1}^{r} \beta_j \mathbf{x}_j$, where the coefficients $\beta_1, ...\beta_r$ are chosen uniformly at random from $\mathbb{F}_q$. Clearly, $\mathbf{x}_{new}$ is also a linear combination of the K original packets. This new combination, along with the coefficients *with respect to the original packets*, is forwarded to node v. If among $\mathbf{x}_1, ..., \mathbf{x}_r$, there is at least one combination that cannot be linearly expressed by the combinations stored in node v, node u has useful (i.e., *innovative*) information for node v, and the $\mathbf{x}_{new}$ is useful to node v (i.e., increases the rank of node v) with probability greater than or equal to $1 - 1/q$ (*Lemma* 2.1 in [11]). If node u knows the encoding matrix of node v (through full signaling that is discussed in Section II-E), it can generate a useful combination using the deterministic algorithm proposed in [21]. Such processing trades off computational overhead for savings in transmission bandwidth, and is not considered in this paper.

RLC schemes incur computational overhead as nodes perform random linear combinations and the destination performs decoding operations. While the complexity of the encoding operation grows linearly with the generation size, the decoding operation has quadratic complexity in the generation size.

E. Design Space

Having reviewed the basic operations of both non-coding and RLC schemes, we now discuss various design options

[2] It is possible for the destination to decode one or more original packets before the matrix reaches full rank. For example this happens if its encoding matrix $\mathbf{A}$ contains one or more row vectors that have exactly one non-zero coefficient.

for DTN routing schemes, which affect both performance and overhead of a particular routing scheme. All of these design options except generation management are applicable to both non-coding and RLC schemes.

Control Signaling. In order to discover neighboring nodes, nodes in the DTNs periodically broadcast and listen to beacon messages in order to discover their neighbors, and exchange information about the packets/coded-packets carried by each other. Such control signaling is useful for nodes to decide *whether to transmit and what information to transmit*. We consider the following different control signaling.

- *Normal Signaling*: In this case signaling is limited to periodic beacon messages in order to discover neighbors. A node only transmits information when it detects at least one neighbor.
- *Full Signaling*: After two nodes discover each other via beaconing, they exchange information about the packets they carry, in order to avoid useless transmissions. In the case of the non-coding scheme, they exchange the sequence numbers of the packets they carry. The node then transmits only those packets that the other does not carry. Similarly, when RLC is employed, the nodes exchange the encoding vectors, so that each node avoids transmitting if it has no useful information for the other node. As a result, under full signaling, a node reaches full rank (i.e., rank K) with probability greater than or equal to $(1-1/q)^{K-1}$ after receiving K encoded-packets of a generation from other nodes.

Simulation results reported in the paper are for full signaling case, unless otherwise specified as in Section III-D2.

Transmission Scheduling and Buffer Management. Routing schemes for DTNs with resource constraints need to deal with resource contention through transmission scheduling and buffer management ([5], [27]). When a node encounters another node, the scheduler decides, among all candidate packets or generations in its buffer, which packets or generations to transmit to the other node. When a node with a full buffer receives a new (encoded) packet, it decides whether and how to make space for the new packet based on its buffer management policy. The schemes proposed [5], [27] estimate the utility of each packet in order to optimize some system performance metric and then select the packet or the combination to transmit or drop accordingly.

In this paper, we consider randomized transmission scheduling for both the non-coding scheme and the RLC scheme. When there are multiple unicast flows in the network, during an encounter, a node gives higher transmission priorities to packets/generations destined to the receiver node; furthermore, among such packets/generations, those originated from the node itself are served first. Under the non-coding scheme, a node selects uniformly at random a packet among candidate relay packets with the same priority, and perform a *round robin scheduling* among source packets it carries [3]. For the RLC scheme, during an encounter, a node selects uniformly at random a generation to transmit from all candidate generations with the same priority. Scheduling among packets from the same generation is performed via random linear coding operation, i.e., a node transmits a random linear combination of all encoded-packets to the other node.

As for buffer management, we consider the *drophead scheme* for non-coding schemes: when buffer is full, the node drops the relay packet that has resided in the buffer the longest. For the RLC scheme, when a node with a full buffer receives a new encoded packet, it chooses a generation from its buffer that has the highest rank (tie is broken randomly). If the newly received packet belongs to the selected generation, each encoded packet of the generation is replaced with its random linear combination with the newly received packet. Otherwise, the node randomly chooses two encoded packets from the chosen generation, and replaces them with their random linear combination.

Recovery Scheme. Multi-copy DTN routing schemes such as epidemic routing and spray-and-wait scheme often employ *recovery schemes* to save resource [15], [49]. For example, under the *VACCINE* recovery, an *anti-packet* (delivery acknowledgement information) is generated by the destination when it first receives a packet, which is then propagated in the entire network, in the same fashion that data packets propagate under epidemic routing, to delete obsolete copies of the packet. We focus on VACCINE recovery as it leads to the most significant resource savings among the different recovery schemes, We extend VACCINE (and any other) recovery scheme to work with RLC so that when a generation of packets is first delivered to its destination, the destination generates an *anti-generation* for this generation, the anti-generation is then propagated in the network to delete remaining copies of packets or encoded packets belonging to the generation.

Replication Control. In resource constrained DTNs where nodes have limited energy, or finite transmission bandwidth, or both, it is beneficial to control the total number of times that a packet (or a generation) is transmitted in the network, through so called *replication control* mechanisms.

For example, under the binary spray-and-wait ([42], [40]), the source node assigns a counter value (a number of *tokens*), denoted as C, to each source packet it generates, which specifies the maximum number of copies that can be made for the packet in the network. When a node carrying a packet with token value $c, (c \geq 2)$ meets another node that does not carry a copy of the packet, the packet is copies to the latter node and the c tokens are equally split between the two copies of the packet, i.e., the former copy keeps $\lceil c/2 \rceil$ tokens and the new copy is assigned $\lfloor c/2 \rfloor$ tokens. A node carrying a packet with token value of $c \leq 1$ can only deliver the packet to the destination. In this way, the total number of copies made for the packet in the whole network is upper-bounded by C, though the actual number of copies being made is often smaller when a recovery scheme is employed. In Section III-C, we propose a replication control scheme based on the binary spray-and-wait to be used in conjunction with RLC.

Generation Management. An RLC scheme needs to address the question of how many and which packets to form a generation. Packets cannot be arbitrarily coded together. First,

[3]This helps to achieve a better balance in the early phase of the dissemination, when small differences in the number of copies of different packets can be amplified by epidemic diffusion.

as we have observed, the overhead of transmitting and storing encoding coefficients grows with the generation size, and so does the complexity of encoding and decoding operations. Second, for unicast applications, when K packets destined to K different nodes are coded together, each of the K destinations has to receive K encoded packets in order to decode the one packet destined to it. We discuss in more depth the impact of generation management in Section IV-A.

III. Single Source Case

In this section, we study the case where a group of packets, from a single unicast source, propagate in a DTN where bandwidth and buffer are constrained.

We first present an algorithm to calculate the *minimum (group) delivery time* under a contact trace and buffer constraint, provide intuition about why RLC schemes (without replication control) achieve this minimum time with higher probability than non-coding schemes, and present a lower bound for this probability (Section III-A). We then discuss the performance of RLC schemes in terms of other metrics (Section III-B), and demonstrate that when replication control is employed, RLC schemes can improve the *delay-per-transmission* in comparison to non-coding schemes (Section III-C). Finally, we discuss how bandwidth and buffer constraints, different control signaling levels and realistic mobility model affect the benefits of RLC schemes (Section III-D).

A. Probability to Achieve Minimum Delivery Time

We use the 4-tuple (s, d, t_0, K) to denote a group of K unicast packets generated by source node s at time t_0, all of which are destined for the same destination d. For (s, d, t_0, K) that can be delivered to the destination under the contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$, there is a *minimum (group) delivery time* by which all the K packets can be delivered to the destination. This time is in general achievable only by a centralized oracle with knowledge of all future contacts. The minimum delivery time clearly lower bounds the delivery time achievable by any routing scheme, and therefore is an ideal benchmark to compare different routing schemes with. We first propose an algorithm for calculating the minimum group delivery time.

1) The algorithm: We first explain how to determine whether the group of K packets can be delivered under the contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$. To address this issue, we first build the event-driven graph $\mathcal{G}(\mathcal{L}, \mathcal{B})$, and then enlarge this graph by adding two nodes: node (s, t_0) that is connected by an intra-node edge with capacity K to the node (s, t_1), where t_1 is the time of the first contact after t_0 involving node s, and a special node (d) to which all nodes involving node d is connected. These edges have a capacity of K, as up to K packets can be transmitted from node (d, t) (with $t > t_0$) to (d). We also change the capacity of all intra-node edges connecting nodes (s, t_1) and (s, t_2) or (d, t_3) and (d, t_4) to K, as we assume nodes have enough buffer to store source packets or packets destined for them. We denote this augmented event-driven graph as $\mathcal{G}'(\mathcal{L}, \mathcal{B}, (s, d, t_0, K))$. For example, Fig. 2 plots the augmented event-driven graph for

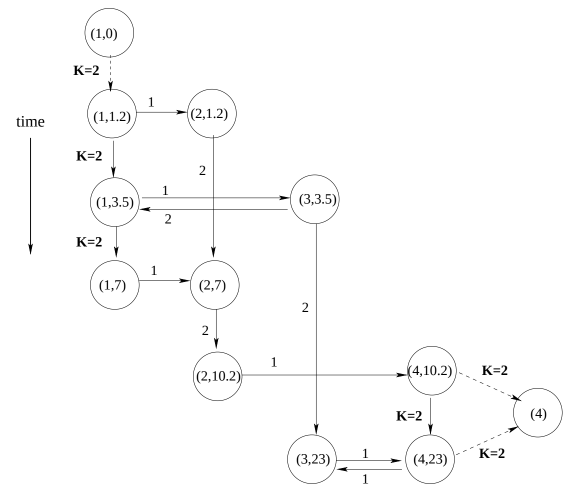


Fig. 2. Augmented event-driven graph $\mathcal{G}'(\mathcal{L}, \mathcal{B}, (1, 4, 0, 2))$ for calculating minimum delivery time for $(1, 4, 0, 2)$, with $\mathcal{B}(u) = 2, u \in \mathcal{V}$. The newly added edges are drawn with dashed lines, and the updated intra-node edge capacity is highlighted using bold font. The maximum flow from $(1, 0)$ to (4) is 2, achieved by the following two paths $(1, 0), (1, 1.2), (2, 1.2), (2, 7), (2, 10.2), (4, 10.2), (4)$, and $(1, 0), (1, 1.2), (1, 3.5), (3, 3.5), (3, 23), (4, 23), (4)$.

the group of packets $(1, 4, 0, 2)$ under the DTN trace depicted in Fig 1, with $\mathcal{B}(u) = 2, u \in \mathcal{V}$.

Based on Proposition 2.1, the group of packets, (s, d, t_0, K), can be delivered under contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$ if and only if there is a flow of value at least K from (s, t_0) to (d) in $\mathcal{G}'(\mathcal{L}, \mathcal{B}, (s, d, t_0, K))$. We therefore have the following proposition:

Proposition 3.1: To determine the minimum delivery time for the group of packets (s, d, t_0, K) under a contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$, it suffices to find the shortest left subsequence of $\mathcal{L}$ (call it $\mathcal{L}_{\min}$) such that the augmented event-driven graph $\mathcal{G}'(\mathcal{L}_{\min}, \mathcal{B}, (s, d, t_0, K))$ can support a flow of value K from (s, t_0) to (d). The time of the last contact in $\mathcal{L}_{\min}$ is the minimum delivery time.

Algorithm MIN_DELIVERY_TIME (Alg.1) intertwines the steps of searching for $\mathcal{L}_{\min}$ with the iterations of the Ford-Fulkerson algorithm for the maximum-flow problem [26]. Starting with an empty augmented event-driven graph $\mathcal{G}_f = \mathcal{G}'(\emptyset, \mathcal{B}, (s, d, t_0, K)) = \langle \{(s, t_0), d\}, \emptyset \rangle$, the algorithm iterates the *expand graph phase* and the *find max-flow phase* until the value of the flow reaches K or all contacts in $\mathcal{L}$ have been processed (in this case the K packets cannot be delivered under the trace).

In the *expand graph phase*, the graph $\mathcal{G}_f$ is expanded by considering new events from L according to their time order, until FIND_PATH $(\mathcal{G}_f, (s, t_0), (d))$ finds a new path with a non-zero residual capacity[4] from node (s, t_0) to node (d). Here GROW$(\mathcal{G}_f, \mathcal{B}, l)$ expands $\mathcal{G}_f$ by processing contact $l \in \mathcal{L}$, following the procedure in Section II-A.

Once a path is found, the algorithm enters the *find max-flow phase* where the flow is augmented until the max-flow from node (s, t_0) to (d) in $\mathcal{G}_f$ is determined. The

[4] The residual capacity of an edge is the difference between its capacity and its current flow value, i.e. how much the flow can still be increased on that link. The residual capacity of a path is defined as the minimum of the residual capacities of all edges in the path.

Ford-Fulkerson algorithm [26] is used for finding the maximum-flow. While this is not the most efficient max-flow algorithm, it allows us to incrementally augment the flow instead of start the maximum flow calculation from scratch, every time the graph is expanded. The procedure UPDATE_RESIDUAL_GRAPH($\mathcal{G}_f$,P) implements the following two steps of Ford-Fulkerson algorithm: augmenting the flow along path P and updating the residual graph. The value b is the increment of flow value due to path P.

If a flow of value at least K is determined, the algorithm returns the time of the last contact that has been considered. Otherwise, it returns a negative value $-f$ (the sign denotes the failure to deliver the whole group of packets, while the absolute value denotes the number of packets that can be delivered). Let $\mathcal{L}'$ be the subsequence of the contact trace $\mathcal{L}$ considered up to termination, the computational complexity of Alg. 1 is $O(K|\mathcal{L}'|)$.

Algorithm 1 MIN_DELIVERY_TIME ($\mathcal{L}, \mathcal{B}, s, d, t_0, K$), find minimum delivery time for the group of packets, (s, d, t_0, K), under contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$

Input: $\mathcal{L}, s, d, K$
$\mathcal{L}_r = \mathcal{L}$, $f = 0$, $\mathcal{G}_f = \langle\{(s, t_0), (d)\}, \emptyset\rangle$
while $f < K$ and $\mathcal{L}_r \neq \emptyset$ **do**
    // *Expand Graph Phase*
    repeat
        // Expand graph until a contact to node d is found
        repeat
            l =pop($\mathcal{L}_r$) // Extract next contact from $\mathcal{L}_r$
            $\mathcal{G}'_f$ =GROW($\mathcal{G}_f, l, \mathcal{B}$), $\mathcal{G}_f \leftarrow \mathcal{G}'_f$
        until $r(l) = d$ // Until the node d is the receiving node of contact l
        P =FIND_PATH($\mathcal{G}_f, (s, t_0), (d)$)
    until $P \neq null$
    // *Find Max-Flow Phase*
    while $P \neq null$ and $f < K$ **do**
        ($\mathcal{G}'_f, b$)=UPDATE_RESIDUAL_GRAPH($\mathcal{G}_f, P$)
        $\mathcal{G}_f \leftarrow \mathcal{G}'_f$, $f \leftarrow f + b$
        P =FIND_PATH($\mathcal{G}_f, (s, t_0), (d)$)
    end while
end while
if $f \geq K$ **then**
    return $t(l)$ // return the time of contact l
else
    return -f // return the negative of f
end if

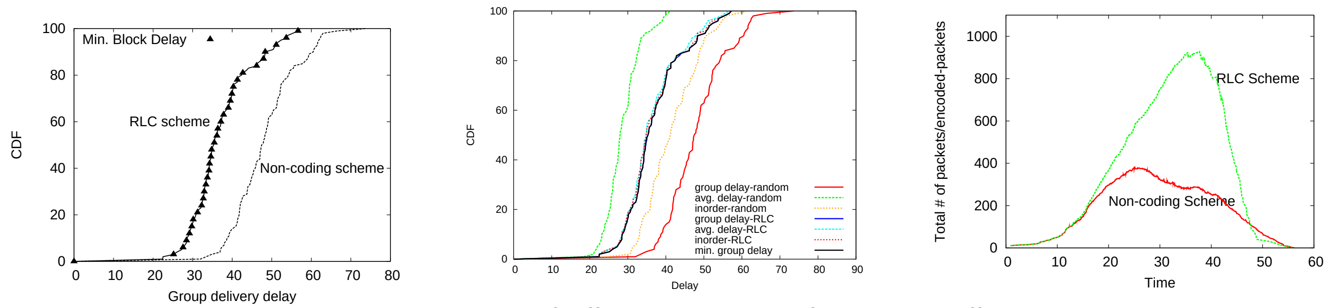
Alg.1 can be easily extended to return the set of paths that supports the flow of value K in the event-driven graph. The set of paths corresponds to a specific DTN routing schedule that achieves the minimum delivery time under the DTN contact trace. For example, the set of two paths $(1, 0), (1, 1.2), (2, 1.2), (2, 7), (2, 10.2), (4, 10.2), (4)$ and $(1, 0), (1, 1.2), (1, 3.5), (3, 3.5), (3, 23), (4, 23), (4)$ in Fig 2, which supports a flow of value 2 from $(1, 0)$ to node (4), corresponds to a set of K forwarding paths in the DTN that achieves the minimum delivery time, 23, for the group of packets $(1, 4, 0, 2)$.

2) Probability to achieve minimum delivery time: In practical settings, network nodes, without prior knowledge about contacts in the network, might choose "wrong" packet(s) (or encoded packet(s) for RLC schemes) to forward during a contact or to delete when the buffer is full. As a result, the destination might receive redundant information through the K (minimum group delivery time) forwarding paths, and more time is needed to deliver the group of packets. In comparison to non-coding schemes, RLC schemes reduce the probability of making wrong choices, due to the larger set of possible useful encoded packets: at a given time, the number of linear combinations useful for the destination is much greater than the number of useful packets. For example, under a randomized non-coding scheme, if a relay node carries $r \leq K$ packets, one of which has already been delivered to the destination, the probability that this relay chooses to forward the useless packet is $1/r$. Whereas under the RLC scheme, if the rank of a relay node is r, and the destination carries one combination that can be linearly expressed by the r encoded packets carried by this relay node, the probability that the random linear combination forwarded by the relay node is useless for the destination is $1/q^{r-1}$ where q is the size of the finite field. In general, $r > 2$, $1/q^{r-1} << 1/r$ (e.g., $q = 2^8$ is a commonly used finite field size in RLC).

Let η be the number of transmission scheduling and buffer management decisions that network nodes make under the RLC scheme *along the set of K forwarding paths that achieves the minimal delivery time*. We know η is upper bounded by the total hop count (including intra-node edges and inter-node edges) of the set of paths that supports the flow of value K in the event-driven graph. Using Fig. 2 as an example, along the set of two forwarding paths that achieves the minimal group delivery time, network nodes need to make three transmission scheduling decisions, respectively during the contacts $(1, 2, 1.2)$, $(1, 3, 3.5)$ and $(2, 4, 10.2)$ [5]. Note that the transmission during contact $(3, 4, 23)$ does not involve scheduling decision as node 3 has only one candidate packet in its buffer at time $t = 23$. As the buffer is not constrained (i.e., $\mathcal{B}(\cdot) = K$) in this example, the number of buffer management decisions is zero. Therefore we have $\eta = 3$. Making use of the correspondence between the RLC based DTN routing scheme and RLC based routing scheme in static network, and Theorem 3 in [18], we prove the following proposition that provides a lower bound on the probability that the RLC scheme achieves minimum group delivery delay. For the outline of the proof, please refer to Appendix A.

Proposition 3.2: Consider a group of packets (s, d, t_0, K) propagating under a contact trace $\mathcal{L}$ with buffer constraint $\mathcal{B}(\cdot)$, and a set of K forwarding paths that achieves the minimum group delivery time. Let η be the number of scheduling and buffer management decisions that DTN nodes perform under the RLC scheme along this set of paths. The RLC scheme achieves the minimum group delivery time with probability

[5] When node 1 encounters node 2 and 3 respectively at $t = 1.2$ and $t = 3.5$, it has two candidate packets to transmit, and needs to decide what to transmit to the other node. When node 2 encounters node 4 at $t = 10.2$, it has two encoded packets in its buffer, and needs to decide what to transmit to node 4.



(a) CDF of group delivery delay, 100 runs

(b) CDF of different delay metrics of 100 runs.

(c) Buffer occupancy under one simulation run

Fig. 3. DTN with $N = 101$ nodes, homogeneous exponential inter-contact time with rate $\beta = 0.0049$, bandwidth constraint of $b = 1$ packet per contact, and unlimited buffer space, without replication control

greater than or equal to $(1 - 1/q)^{\eta}$.

Fig. 3(a) plots the empirical cumulative distribution functions (CDFs) of the minimum delivery delay, and the delivery delay achieved by the RLC and the non-coding scheme over 100 different simulation runs. We observe that the RLC scheme does achieve the optimal performance (in terms of group delivery delay) with high probability, whereas under the non-coding scheme, the delivery delay is significantly larger.

B. Other Delay Metrics and Number of Transmissions

We now compare the RLC scheme and non-coding scheme in terms of other delay metrics and the total number of transmissions made in the network.

We first consider the average packet delay and average in-order packet delay. Fig. 3(b) plots the empirical CDFs of different delay metrics achieved by the RLC scheme and the non-coding scheme from 100 different simulation runs. The four almost overlapping curves are CDFs of the minimum group delay and the three different delay metrics achieved by the RLC scheme. Under the RLC scheme, the average delay and the average in-order delay are stochastically slightly smaller than the group delivery delay. In contrast, the non-coding scheme shows significantly different performance: it fares worse in terms of group delivery delay and average in-order packet delay, but better in terms of average packet delay. Note that the RLC scheme considered here has some specific implementation peculiarities that improve its performance in terms of average delivery delay. For example, if a node can decode one or multiple packets before it reaches full rank, it forwards the decoded packet(s) (rather than random linear combinations) to destination.

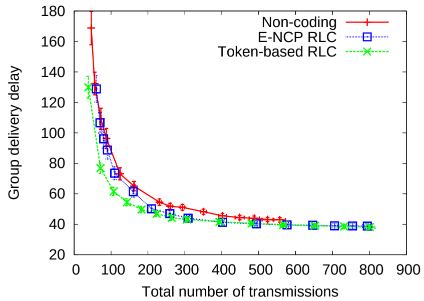
The RLC scheme achieves faster information propagation at the price of a greater number of transmissions and a larger buffer occupancy. For example, Fig. 3(c) plots the total numbers of packet copies (for the non-coding scheme) or combinations (for the RLC scheme) in the network as a function of time for one particular simulation run (the group of packet is generated at time $t = 0$). Due to the increased randomness of RLC, the probability that two nodes that meet each other have useful information to exchange is higher. As a result, we observe a sharper increase in the total number of copies/combinations in the network. Furthermore, under RLC, the recovery process starts only when the whole generation is delivered, whereas under the non-coding scheme, the recovery process for an individual packet starts immediately when the packet is delivered.

In order for the RLC scheme to be beneficial in resource constrained DTNs with competing traffic, the RLC scheme needs to improve the delay performance without incurring higher transmissions overhead than the non-coding scheme. We investigate this in the next section.

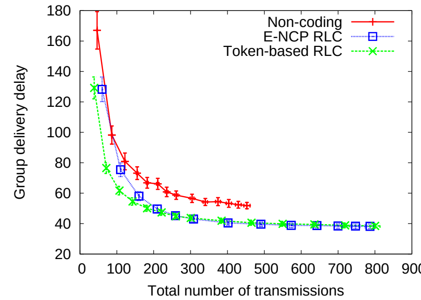
C. Delay vs. Number of Transmissions Trade-off

In order to control the number of transmissions in RLC schemes, we propose the *token-based* RLC scheme which extends the binary spray-and-wait. A certain number of tokens (denoted as C_g) is assigned to each generation to limit the total number of combinations that can be transmitted for this generation in the network. The operation of RLC schemes as described in Section II-D is complemented with the following consideration about tokens. We focus on a particular generation so that we can talk about the number of tokens and the rank of a node without specifying the generation. When two *non-destination* nodes meet, they redistribute their tokens in proportion to their ranks (see [50] for more details). Then each of the two nodes transmits a random linear combination to the other if it has useful information and if it has more than one token. After each transmission, the sending node reduce its number of tokens by one. The two procedures (token reallocation and transmission of one combination) are repeated until the contact terminates. This way, the total number of transmissions made to *non-destination* nodes is bounded by C_g. When a node meets the destination, it transmits as many combinations as it can, independently from its number of tokens. Under full signaling, the total number of transmissions to the destination (for the destination to reach full rank) is K with probability greater than or equal to $(1 - 1/q)^{K-1}$. In summary, this scheme limits the total number of transmissions in the network to $C_g + K$ with high probability. The actual number of transmissions is smaller when a recovery scheme is employed.

A different replication control scheme, called *E-NCP* (Efficient Protocol based on Network Coding), was proposed in [30]. In order to deliver a group of K packets, the source disseminates K' (slightly larger than K) random linear combinations (which are referred to as *pseudo source packets*) to



(a) Delay vs number of transmissions trade-off ($b = 1, B = \infty$)



(b) Delay vs number of transmissions trade-off ($b = 1, B = 2$)

Fig. 4. Group delivery delay vs number of transmissions trade-off achieved when replication control is employed

the first K' relays that it encounters. Each of the K' relays then uses binary spray-and-wait to limit the total number of transmissions made for the pseudo source packet it carries. Different pseudo source packets are randomly linearly combined at intermediate nodes, as under regular RLC scheme. The reason for disseminating K' pseudo source packets is so that the original K source packets can be decoded with high probability when the destination receives K encoded-packets[6].

We compare the group delivery delay versus transmission number trade-off achieved by the non-coding scheme (with binary spray-and-wait applied to each of the K packets), the token-based RLC scheme, and the E-NCP scheme. Fig. 4 plots the average group delivery delay versus the average number of transmissions (together with the 95% confidence intervals for both metrics), for a group of $K = 10$ packets, under different token limits, for the cases both without buffer constraints (a) and with a buffer constraint of $B = 2$ (b). We observe that, with a similar number of transmissions, both RLC schemes achieve smaller group delivery delay than the non-coding scheme. Token-based RLC scheme outperforms E-NCP, especially under small number of transmissions. The results for the limited relay buffer case further establish the benefits of the RLC schemes in reducing group delivery delay without increasing transmission overhead.

D. Discussion of RLC benefits

In this section, we study how different system parameters affect the benefits and overhead of RLC schemes.

1) Impact of Different Bandwidth and Buffer Constraints: We first consider the impact of varying bandwidth constraint while fixing the buffer constraint $B = K$ (i.e., no buffer constraint). We observe that as the network bandwidth becomes less and less constrained, the benefit of RLC diminishes and disappears when the number of packets that can be exchanged during each contact, b, equals the group size K. In this case, the K packets propagate independently without competing for bandwidth, and the group delivery delay coincides with the epidemic routing delay under no resource constraints as characterized in [49]. For example, Fig. 5.(a) plots the average group delivery delay and its 95% confidence interval (based on 50 different simulation runs) under varying bandwidth constraints, for a group of $K = 10$ packets from the same unicast flow.

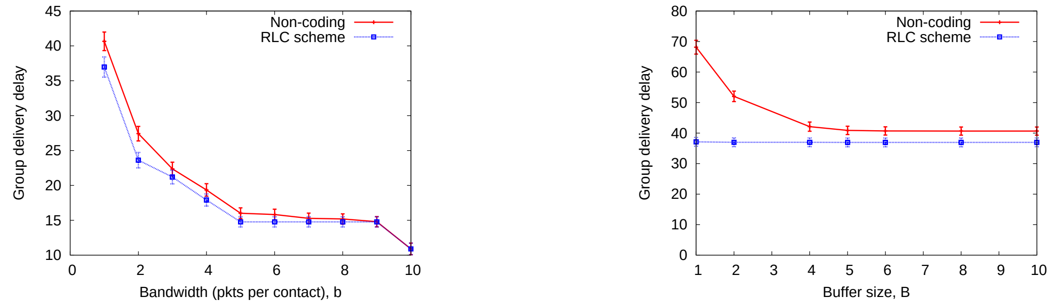
We now consider the benefit of RLC schemes under varying buffer constraint. Fig. 5(b) plots the average group delivery delay (and the 95% confidence interval) for a group of $K = 10$ packets achieved by the RLC scheme and the non-coding scheme under varying nodal buffer sizes, B. We observe that as buffer space becomes more and more constrained, performance under the RLC scheme degrades only slightly, in sharp contrast to the non-coding scheme. As different packets are mixed randomly by nodes under the RLC scheme during transmission or buffer management decision, the RLC scheme allows a more uniform distribution of different packets in the network. For the non-coding scheme, the more copies a packet has in the network, the more the packet is copied to other nodes and evicted a copy of another packet when buffer is full[7]. This results in an uneven propagation of different packets: some packets spread quickly to a large number of nodes, while others spread much more slowly. It therefore takes much longer to deliver the "slowest" packet and therefore the whole group of packets.

2) Impact of Control Signaling: The simulation results presented so far are for the *full signaling* case, where two encountering nodes first exchange information about which packets or encoded packets they carry, and decide whether and what packets to transmit to the other node based on this information. Full signaling incurs a higher transmission and computational overhead for the RLC scheme than for the non-coding scheme, as each node needs to exchange the encoding matrix (in comparison to packet IDs), and determine whether it has useful information for the other based on the received encoding matrix.

We now consider *normal signaling*, where two nodes encountering each other do not exchange information about the packets or encoded packets they carry. For the non-coding scheme, a node randomly chooses a packet from the set of packets it carries, and forwards it to the other node; for the RLC scheme, a node always generates and transmits a

[6] We observe that it would be sufficient for the source to disseminate the original K source packets or K linearly independent encoded-packets to K relay nodes.

[7] As a first approximation, the spreading rate of a packet with n copies in a network with N nodes is proportional to $n(N - n)$. Therefore, a packet with a larger number of copies spreads faster as long as $n < N/2$.



(a) Group delivery delay under varying bandwidth

(b) Group delivery delay under varying buffer, $b = 1$

Fig. 5. Impact of bandwidth and buffer constraints

random linear combination to the other node. We observe from the simulation results that, while the performance of the non-coding scheme is significantly affected by the lack of information, the RLC scheme is almost not affected. In particular, Fig. 6 plots the group delivery delay versus the number of transmissions trade-off achieved by the non-coding and the RLC scheme under full signaling and normal signaling respectively. We observe that the performance of the RLC scheme under normal signaling is almost identical to that under full signaling, whereas for the non-coding scheme, the performance under normal signaling is significantly worse.

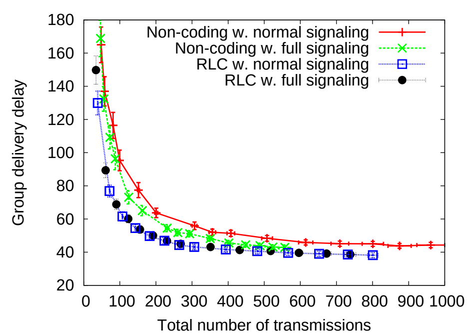


Fig. 6. Group delivery delay vs number of transmissions trade-off under full signaling and normal signaling

3) Impact of Real Mobility Behavior: To study the impact of real mobility, we compare the performance of the RLC scheme and non-coding scheme using contact traces collected from the UMass DieselNet [7] testbed in the spring semester of year 2006. Our experiments are described in [50] and support the previous findings. As the DieselNet contact traces correspond to a very challenging scenario where most of the packets cannot be delivered on a time horizon of 12 hours, the RLC performance improvement appears in terms of an increased delivery probability in comparison to non-coding schemes (respectively 31% and 24%).

IV. Multiple Unicast Flows

We have shown that RLC schemes achieve faster delivery of a group of packets from the same unicast flow than non-coding schemes, at the cost of a larger number of network transmissions. Furthermore, when replication control is employed, RLC schemes improve the trade-off between delivery delay and transmission number.

The next question to ask is whether RLC schemes provide any benefit when multiple unicast flows are present in the network. The presence of multiple flows adds a new dimension to generation management, in fact one can limit coding to packets belonging to the same flow (*intra-flow* coding), or allow coding packets belonging to different flows (*inter-flow* coding), where nodes combine packets from different sources but destined for the same destination, or even combine packets regardless of their source and destination.

Next, we first examine the benefits achieved by RLC under inter-flow coding for the case where there is a single generation in the network. We then study intra-flow coding in a network with multiple unicast flows.

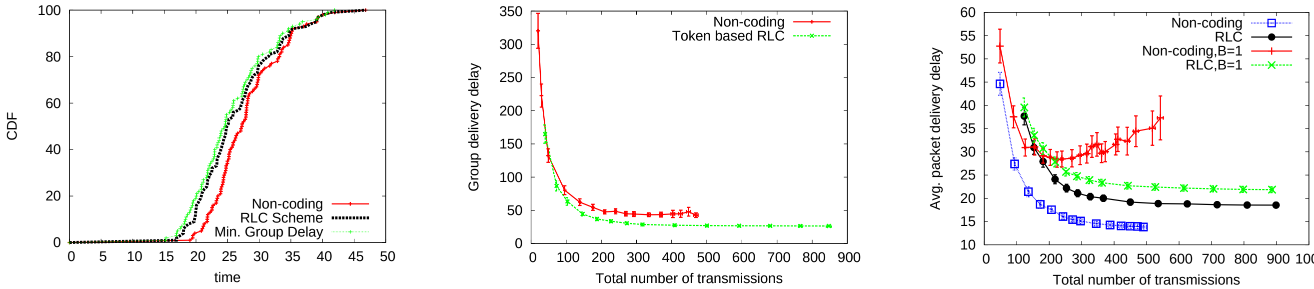
A. Inter-Flow Coding

The focus of Section III is the benefit of RLC when applied to a group of packets originating from a single source and destined for a single destination, i.e., the Single-Source Single-Destination (*SS_SD*) case. Now we investigate the benefit of applying RLC to:

i) a group of K packets originating from K different sources and destined for the same destination, i.e., Multiple-Sources Single-Destination (*MS_SD*) case, and

ii) a group of K packets originating from K different sources and destined for K different destinations, i.e., Multiple-Sources Multiple-Destinations (*MS_MD*) case. We assume that each of the K destination nodes also act as relay for packets destined for other nodes.

For the *MS_SD* case, Alg 1 can be extended to calculate the minimum group delivery time (details are given in [50]). We perform simulation to compare the group delivery delay achieved by the RLC scheme and the non-coding scheme against this baseline, and plot the CDFs (from 100 different simulation runs) of the minimum delivery delay, the group delivery delays under the non-coding and RLC scheme in Fig 7(a). We note that the RLC scheme achieves smaller group delivery delays than the non-coding scheme, and the delays are close to the minimum possible. We also observe that the reduction in group delivery delay achieved in this case is much smaller than that of *SS_SD* case (Fig 3(a)). The explanation is that, under the MS_SD case, the K packets start to propagate from K different source nodes, therefore, the contention among the K packets starts later in time, and



(a) MS_SD: Empirical CDF of group delivery delay

(b) MS_SD: Group delivery delay vs. number of transmissions trade-off, $B = 2$

(c) MS_MD: avg. packet delivery delay vs number of transmissions trade-off

Fig. 7. Benefit of RLC coding under inter-flow coding

the effect of relay nodes under non-coding schemes choosing wrong packets to forward becomes less significant.

For the MS_MD case, determining the minimum group delivery time under a contact trace (and buffer constraint) is a NP-hard problem[8]. Furthermore, as each of the K packets is destined for a different destination, it is more meaningful to consider the *average* time for each of the destinations to receive the one packet destined for itself (i.e., *average packet delivery delay*), than the time to deliver the last packet in the group (i.e., *group delivery delay*). We therefore use the *average packet delivery delay* as the performance metric for the MS_MD case.

We now investigate whether inter-flow RLC coding can improve the delivery delay versus number of transmissions trade-off through simulation studies. First, we extend the token scheme to the inter-flow coding. When applying the token scheme to the *MS_SD* and *MS_MD* cases, a per-packet token number C is assigned to each of the K packets by its respective source upon packet generation. As with the *SS_SD* case, a node is always allowed to transmit to the destination (for the *MS_SD* case) or one of the K destination nodes (for the *MS_MD* case), even when its token number is zero. Similar to the *SS_SD* case, the total number of transmissions made to non-destination nodes is bounded by CK, and with high probability, a destination reaches full rank and stop receiving transmissions after receiving K combinations. Therefore, with high probability, the total number of transmissions made in the network is bounded by $CK + K$ under the *MS_SD* case, and is bounded by $CK + K^2$ under the *MS_MD* case.

Simulation studies show that for the *MS_SD* case, the RLC scheme and the non-coding scheme achieve almost identical trade-off curves when buffers are not constrained. However, when the buffer is constrained, the RLC scheme improves the trade-off, as illustrated in Fig. 7(b). For the *MS_MD* case, we compare the *average packet delivery delay* versus the total number of transmissions made trade-off achieved by the non-coding and the RLC scheme. Fig. 7(c) plots the results for both *i)* the case where only bandwidth is constrained ($b = 1$), and *ii)* the case where both bandwidth and buffer are constrained ($b = 1, B = 1$). We observe that the RLC scheme performs worse than the non-coding scheme for the former case. This is reasonable as the RLC scheme forces each destination to receive K independent combinations in order to decode the one single packet destined for it. When buffers are also constrained, we observe that with a small total number of transmissions, the RLC scheme performs worse than the non-coding scheme; however, when a relatively larger number of transmissions is allowed, the RLC scheme achieves better trade-off than the non-coding scheme.

Our discussion in this section demonstrates that RLC is most advantageous when it is applied to packets from the same flow. In the next section, we focus on applying intra-flow coding to the case of multiple continuous flows.

B. Multiple Continuous Flows with Intra-flow coding

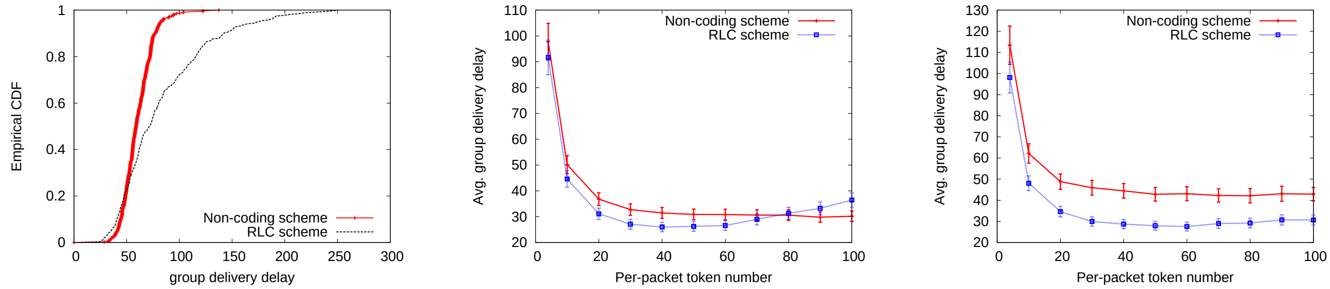
We assume there are N unicast flows in the network, and each source independently generates groups of $K = 10$ packets according to a Poisson process with rate λ. RLC is applied to packets belonging to the same group.

We perform simulation studies for a network with $N = 101$ nodes, bandwidth constraint $b = 1$. We compare the average group delivery delay under the RLC scheme and the non-coding scheme without replication control. We observe that the RLC scheme only exhibits a benefit when the traffic rate is low; and performs worse than the non-coding scheme when the traffic rate is high. For example, Fig 8(a) shows that it is not beneficial to apply RLC in our reference scenario for a rate $\lambda = 0.45 \times 10^{-3}$, a relatively high traffic rate.

We can explain this result as follows. First, at a relatively high traffic rate, there is a large number of different packets in the network. As a result, under the non-coding scheme, it is more likely that two nodes can exchange useful information when they meet, and therefore, the RLC scheme achieves a smaller relative benefit over the non-coding scheme through its increased randomization. Secondly, as we have shown in Fig 3(b), RLC schemes incur a larger number of transmissions for each generation than non-coding schemes. Therefore, when the group arrival rate is high, contention for bandwidth under RLC schemes is greater than under non-coding schemes and some of the flows can be severely penalized[9].

[8]The sub-problem of deciding whether the group of packets can be delivered under a contact trace is a form of Edge-Disjoint Paths problem which is known to be NP-hard [10].

[9] Flows with a larger number of combinations in the network are propagated more and then get even more resources. The mechanism is similar to that described in Sec. III-D for non-coding schemes.



(a) CDF of group delivery delay without replication control, $B = \infty$

(b) Group delivery delay under replication control with varying token number, $B = \infty$

(c) Group delivery delay under replication control with varying token number, $B = 3$

Fig. 8. Group delivery delay under multiple generation case, $\lambda = 0.45 \times 10^{-3}$, $b = 1$

To alleviate resource contention, we resort to replication control, which has been shown to allow the RLC scheme to achieve a better delay-transmission trade-off than the non-coding scheme. For both the RLC scheme and the non-coding scheme, we vary the per-packet token limit, C, between 20 and 100. Fig 8(b) plots the average group delivery delay under different per-packet token limits. We observe that the RLC scheme achieves a smaller average group delivery delay only when the token limit is carefully tuned. In particular, for the RLC scheme in this setting, there is an optimal token limit value between 40 and 50: If the token limit is too large, severe contention leads to degraded performance; if it is too small, some contacts are not exploited because all the tokens have been consumed. For the non-coding scheme, for a number of tokens smaller than 100, contention is not significant and a smaller token limit leads to a larger average block delay. We do observe that under a higher traffic rate, the non-coding scheme also benefits from replication control. How to configure replication control schemes (e.g., setting token limit) for a given network setting is an open problem ([51]), and beyond the scope of this paper.

As with the single generation case, the RLC benefit under multiple flow case is more significant when both bandwidth and buffer are constrained. We repeat the simulation as shown in Fig 8(b), introducing buffer constraint of $B = 3$. The result (Fig 8(c)) again shows that RLC is more beneficial when both buffer and bandwidth are constrained. In this particular setting, RLC reduces the average group delivery delay by more than 25% for token values ranging from 20 to 100.

V. Related Work

Several works ([22], [44]) have studied the application of erasure coding ([38], [33], [32]) to DTNs, where the source encodes a message into a large number of blocks, such that as long as a certain fraction or more of the coded blocks are received, the message can be decoded. For DTNs where there is prior knowledge about paths and their loss behavior, Jain *et al.* ([22]) studied how to allocate the coded blocks to the multiple lossy paths in order to maximize the message delivery probability. To reduce the variance of delivery delay in DTNs with unpredictable mobility, Wang *et al.* ([44]) proposed to encode each message into a large number of coded blocks which are then transmitted to a large number of relays helping to deliver the coded blocks to the destination. We note that network coding is a generalization of erasure coding, and the benefits of erasure coding scheme can also be achieved by RLC schemes.

While Widmer *et al.* ([45], [46]) studied the benefit of RLC for broadcast applications in DTNs, we study unicast applications for which replication control and recovery schemes are introduced. Our findings that under normal signaling, the relative benefit of RLC is much more significant than that under full signaling is in line with the similar finding about broadcast application in [45].

We compare our token scheme with the E-NCP scheme proposed in [30] in Section III-C. Using the connection between E-NCP and the low-density distributed erasure codes [3], [30] proved that in order for the destination to decode all K packets with any K encoded packets with high probability, it suffices to set the per-packet token limit to $\Theta(logK)$. In contrast, we compare different replication control schemes in terms of the fundamental performance trade-off between delivery delay and number of transmissions.

Based on several simplifying assumptions[10], Lin *et al.* ([31]) derived ODE models for analyzing delivery delay under the RLC non-coding scheme, for the case of a single group of K packets. We note that due to these simplifying assumptions, the ODE models not only underestimated the delivery delay for both the RLC scheme and the non-coding scheme, bus also underestimated the performance difference between the two schemes.

The benefit of RLC observed in our setting is similar in spirit to that of rumor mongering ([11], [6]). For a network with the so called *random phone call* communication model, where at each time step, each node communicates with another node selected uniformly at random among all the nodes, [11], [6] derived asymptotic bounds for the tim to disseminate multiple messages in the network under both RLC and non-coding schemes. As both the communication model and the schemes considered (no signaling) therein differ from ours, similar analysis does not apply to our setting.

Finally, [29] presented a preliminary investigation on the effect of topology on the RLC performance. Simulation results

[10]For the RLC scheme, it is assumed that an equal fraction of nodes in the network has a rank of $1, \ldots, B-1$(recall B is the number of packets a node can store); for the non-coding scheme, it is assumed that the K packets are equally likely to reside at each node.

for different graphs (Erdös-Rényi, Random Geometric graph, grid, Watts-Strogatz) and the case where there is a single unicast flow in the network were presented.

VI. Conclusions

In this paper we investigate the benefits of applying random linear coding to unicast applications in resource constrained DTNs. Due to its frequent network disconnection and rapidly changing topology, the key challenge for unicast routing in DTNs is distributed packet transmission scheduling and buffer management. Under RLC schemes, each node generates and forwards a random linear combination of stored packets during a contact, and randomly combine newly received packet with existing ones when the buffer is full. Because of its higher degree of randomness compared to non-coding schemes, RLC schemes increase the probability that a node forwards/keeps information useful for the eventual delivery to the destination.

More specifically, from the case of a single group of packets (SS_SD) propagating in the network, RLC reduces the group delivery delay, and achieves minimum group delay with probability greater than or equal to $(1-1/q)^{\eta}$. Larger gains are achieved by RLC schemes when resource (bandwidth and buffer space) is severely constrained, when full information about the content of other nodes is not available (i.e., under normal signaling), and when coding is applied to packets from same unicast flows.

Even though RLC schemes reduce group delivery delay at the price of a larger number of network transmissions, with replication control, RLC improves the trade-off between delivery delay and total number of transmissions. This improved performance trade-off allow RLC schemes to reduce average block delivery delay under multiple continuous unicast flows, with significant performance improvement when node buffer is constrained.

References

[1] First Mile Solutions. http://firstmilesolutions.com/.

[2] Wizzy digital courier. http://www.wizzy.org.za/.

[3] A. G. Dimakis and V. Prabhakaran and K. Ramchandran. Decentralized Erasure Codes for Distributed Networked Storage. *IEEE Transactions on Information Theory*, 52(6):2809–2816, 2006.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. on Information Theory*, 46:1204–1216, July 2000.

[5] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *ACM Conference on Applications, Technologies, and Protocols for Computer Communication (SIGCOMM)*, pages 373–384, 2007.

[6] M. Borokhovich, C. Avin, and Z. Lotker. Tight Bounds for Algebraic Gossip on Graphs. In *IEEE International Symposium on Information Theory (ISIT)*, 2010.

[7] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[8] H. Cai and D. Y. Eun. Crossing over the bounded domain: from exponential to power-law intermeeting time in mobile ad hoc networks. *IEEE/ACM Transaction on Networking*, 17.

[9] H. Cai and E. Y. Eun. Toward stochastic anatomy of inter-meeting time distribution under general mobility models. In *ACM international symposium on Mobile ad hoc networking and computing(MobiHoc)*.

[10] C. Chekuri, S. Khanna, and F. B. Shepherd. An $O(\sqrt{(n)})$ Approximation and Integrality Gap for Disjount Paths and Unsplittable Flow. *Theory of Computing*, 2, 2006.

[11] S. Deb, M. Medard, and C. Choute. Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering. *IEEE/ACM Transactions on Networking, special issue on networking and information theory*, pages 2486–2507, 2006.

[12] A. Doria, M. Ud'en, and D. P. Pandey. Providing Connectivity to the Saami Normadic Community. In *International Conference on Open Collaborative Design for Sustainable Innovation (dyd02)*, Dec 2002.

[13] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network Coding: An Instant Primer. *ACM SIGCOMM Computer Communication Review*, 36, 2006.

[14] R. Groenevelt, P. Nain, and G. Koole. The Message Delay in Mobile Ad Hoc Networks. In *Performance*, October 2005.

[15] Z. J. Haas and T. Small. A New Networking Model for Biological Applications of Ad Hoc Sensor Networks. *IEEE/ACM Transactions on Networking*, 14:27–40, February 2006.

[16] D. Hay and P. Giaccone. Optimal routing and scheduling for deterministic delay tolerant networks. In *WONS 2009 (International Conference on Wireless On-Demand Network Systems and Services)*, pages 27–34, 2009.

[17] T. Ho, R. Koetter, M. Medard, D.R. Karger, and M. Effros. The Benefits of Coding Over Routing in a Randomized Setting. In *IEEE International Symposium on Information Theory (ISIT)*, 2003.

[18] T. Ho, M. Medard, R. Koetter, and D.R. Karger. A Random Linear Network Coding Approach to Multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, October 2006.

[19] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot. Pocket Switched Networking: Challenges, Feasibility, and Implementation Issues. In *IFIP TC6 International Workshop on Autonomic Communication (WAC)*, 2005.

[20] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2006.

[21] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M.G.M. Tolhuizen. Polynomial Time Algorithms for Multicast Network Code Construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.

[22] S. Jain, M. Demmer, R. Patra, and K. Fall. Using Redundancy to Cope with Failures in a Delay Tolerant Network. In *ACM SIGCOMM (Conference on Applications, Technologies, and Protocols for Computer Communication)*, 2005.

[23] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.

[24] T. Karagiannis, J-Y. Le Boudec, and M. Vojnovic. Power Law and Exponential Decay of Inter Contact Times between Mobile Devices. In *ACM Conference on Mobile Computing and Networking (MOBICOM)*, 2007.

[25] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and Inference Problems for Temporal Networks. In *Journal of Computer and System Sciences, Special issue on STOC*, volume 64, 2002.

[26] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, united states ed edition, March 2005.

[27] A. Krifa, C. Barakat, and T. Spyropoulos. Optimal buffer management policies for delay tolerant networks. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 16–20, 2007.

[28] R Lidl and H. Niederreiter. *Finite Fields, 2nd edition*. Cambridge, England: Cambridge University Press, 1997.

[29] L. Lima, D. Ferreira, and J. Barros. Topology Matters in Network Coding. *Springer Telecommunication Systems Journal, Special Issue on Future Internet Architectures*, 52(3):1–11, March 2011.

[30] Y. Lin, B. Li, and B. Liang. Efficient network coded data transmissions in disruption tolerant networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[31] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. *IEEE Journal on Selected Areas in Communications*, pages 794–808, 2008.

[32] M. Luby. LT codes. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 271–282, 2002.

[33] M. Luby, M. Mitzenmacher, A. Shokrollabi, and D. Spielman. Efficient Erasure Correcting Codes. *IEEE Transactions on Information Theory*, February 2001.

[34] D. S. Lun, M. Medard, T. Ho, and R. Koetter. Network Coding with a Cost Criterion. In *International Symposium on Information Theory and its Applications (ISITA)*, 2004.
[35] A. Maffei, K. Fall, and D. Chayes. Ocean Instrument Internet. In *Proc. ASU Ocean Science Conference*, 2006.
[36] J. Partan, J. Kurose, and B. N. Levine. A Survey of Practical Issues in Underwater Networks. In *ACM International Workshop on UnderWater Networks (WUWNet)*, 2006.
[37] M. Pitkanen, A. Keranen, and J. Ott. Message Fragmentation in Opportunistic DTNs. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2008.
[38] I.S. Reed and G. Solomon. Polynomial Codes over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8:300–304, 1960.
[39] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
[40] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient Routing in Intermittently Connected Mobile Networks: The Multi-copy Case. In *ACM/IEEE Transactions on Networking*, volume 16, pages 130–143, 2007.
[41] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient Routing in Intermittently Connected Mobile Networks: The Single-copy Case. In *ACM/IEEE Transactions on Networking*, volume 16, pages 63–76, 2007.
[42] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
[43] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, April 2000.
[44] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
[45] J. Widmer and J.-Y. Le Boudec. Network Coding for Efficient Communication in Extreme Networks. *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
[46] J. Widmer, C. Fragouli, and J.-Y. Le Boudec. Energy-efficient broadcasting in wireless ad-hoc networks. In *IEEE Workshop on Network Coding, Theory, and Applications (NETCOD)*, 2005.
[47] Y. Wu, P. A. Chou, and S.-Y. Kung. Minimum-Energy Multicast in Mobile Ad hoc Networks using Network Coding. In *IEEE Information Theory Workshop*, 2004.
[48] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks. In *IEEE Workshop on Network Coding, Theory, and Applications (NETCOD)*, 2006.
[49] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance Modeling of Epidemic Routing. *Elsevier Computer Networks journal*, 51/10:2859–2891, 2007.
[50] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang. Random Linear Coding for Unicast Applications in Disruption Tolerant Networks. Technical Report 7277, INRIA, 2010. http://hal.archives-ouvertes.fr/inria-00494473/en/.
[51] X. Zhang, H. Zhang, and Y. Gu. Impact of Source Counter on Routing Performance in Resource Constrained DTNs. *Elsevier Journal of Pervasive and Mobile Computing*, 7(2):172–188, 2010.

APPENDIX

Proof of Proposition 3.2: The propagation of a group of packets (s, d, t_0, K) under a DTN trace $\mathcal{L}$ with buffer constraint $\mathcal{B}(\cdot)$ by a RLC scheme corresponds to the propagation of K packets from node (s, t_0) to node (d) under a RLC scheme in $\mathcal{G}'(\mathcal{L}, \mathcal{B}, (s, d, t_0, K))$, where information transmitted over each edge is a linear combination of the edge's input. More specifically, without loss of generality, we assume each edge in $\mathcal{G}'$ has a unit capacity [11], and denote the origin node of (directed) edge e as $o(e)$, and the destination node of edge e as $d(e)$. We have:

i) The K packets originated at node (s, t_0) can be viewed as K independent input processes, each with entropy rate of one packet per unit time.

ii) The information process transmitted on edge e, $\mathcal{Y}_e$, is formed as a linear combination of the edge's inputs. If edge e is an intra-node edge with the source (s, t_0) being its origin, i.e., $o(e) = (s, t_0)$, then Y_e is a linear combination fo the K source packets:

$$\mathcal{Y}_e = \sum_{i=1}^{K} a_{i,e} \mathbf{m}_i \tag{1}$$

Otherwise, $\mathcal{Y}_e$ is a linear combination of all input processes to edge e (i.e., information processes over edge e' whose destination is the origin of edge e):

$$\mathcal{Y}_e = \sum_{e':d(e')=o(e)} f_{e',e} Y_{e'} \tag{2}$$

Note that the linear coding coefficients on some edges are fixed. When buffer is not constrained, the coding coefficients on intra-node edges are fixed. The transmission on some inter-node edges also do not involve random linear coding, as the number of inputs to its origin node is 1. On such edge, the (encoded) packet is simply stored or forwarded as they are, which can be represented as a linear combination with exactly a 1 in its coding coefficient (the rest coefficients are 0). For the rest edges in $\mathcal{G}'$, the coding coefficients are randomly chosen from $\mathbb{F}_q$. We have:

Lemma 1: If the group of packet (s, d, t_0, K) can be delivered under contact trace $\mathcal{L}$ and buffer constraint $\mathcal{B}(.)$, then from node (s, t_0) to node (d) can be attained in the DTN RLC scheme achieves the minimum group delivery delay for (s, d, t_0, K), trace $\mathcal{L}$ with buffer constraint $\mathcal{B}$, if and only if, the corresponding RLC scheme achieves the flow value of K in $\mathcal{G}'(\mathcal{L}_{min}, \mathcal{B}, (s, d, t_0, K)))$, where $\mathcal{L}_{min}$ is the left subsequence of $\mathcal{L}$ with contacts occured up to the minimum group delivery time.

We note that the graph $\mathcal{G}'$ is a static, acyclic, and delay-free directed graph, the same network model as that considered in [18], and the single unicast flow case is a special case of the *multicast connection problem* as studied in [18] with a single source and a single destination. Also, for the edges in $\mathcal{G}'$ where the coding coefficients are fixed, the fixed coefficients are always the same as the coefficients in the solution that attains the flow of value K in $\mathcal{G}'$. Applying Theorem 3 in [18], we conclude that the RLC scheme attains the flow of value K in $\mathcal{G}'$ with a probability greater than or equal to $(1 - 1/q)^{\eta}$. From Lemma 1, we conclude that the same lower bound holds for the probability that the RLC scheme achieves the minimum group delivery delay.

■

[11] As each edge has an integral capacity b, we can replace it with b edges (with same origin and desitnation) with unit capacity.