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 applications in Disruption Tolerant Networks (DTNs). Under RLC, nodes store and forward random linear combinations of packets as they encounter each other. For the case of a single group of packets originating from the same source and destined for the same destination, we prove a lower bound on the probability that the RLC scheme achieves the minimum time to deliver the group of packets. Although RLC significantly reduces group delivery delays, it fares worse in terms of average packet delivery delay and network transmissions. When replication control is employed, RLC schemes reduce group delivery delays without increasing the number of transmissions. In general, the benefits achieved by RLC are more significant under stringent resource (bandwidth and buffer) constraints, limited signaling, highly dynamic networks, and when applied to packets in the same flow. For more practical settings with multiple continuous flows in the network, we show the importance of deploying RLC schemes with a carefully tuned replication control in order to achieve reduction in average delay, which is observed to be as large as 20% when buffer space is constrained.**

I. Introduction

In recent years, wireless communication technologies have been increasingly deployed in environments where there are no communication infrastructures, as evidenced by the many efforts to build and deploy wireless sensor networks for wildlife tracking ([21], [13]), underwater sensor networks ([35], [36]), disaster relief team networks, networks for remote areas or for rural areas in developing countries ([1], [10]), vehicular networks ([6], [18]) and Pocket-Switched Networks ([17]). 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 other 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, DTNs often face 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, neither buffer space nor power are severely constrained, but transmission bandwidth is still a scarce resource. To address these challenges, a plethora of routing schemes have been proposed for DTNs ([43], [42], [39], [13], [40], [41], [4]): some of these works explore the trade-off between routing performance and resource consumption, whereas others attempt to optimize routing performance under certain resource constraints.

Ahlswede *et al.* ([3]) 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. For example, the results obtained in [34], [47] for multicast in static wireless networks are not directly applicable to DTNs due to their dynamically changing topology. Also, in DTNs, each node usually has 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)([48], [23], [32]) 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 ([11]).

Existing research has studied the application of Random Linear Coding (RLC), a special form of network coding [15], to broadcast and unicast communication in DTNs. 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 applications, Widmer *et al.* ([45], [46]) showed that the RLC

Xiaolan Zhang is with the Department of Computer and Information Science, Fordham University, Bronx, NY 10458. Haixiang Wang was with the Department of Computer and Information Science, Fordham University when this work was done. 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.

scheme achieves higher packet delivery rates than the non-coding scheme with the same forwarding overhead. For unicast applications, our earlier work [49] first investigated the benefit of RLC through simulation studies. Lin *et al.* proposed and analyzed a replication control scheme ([30]) for RLC schemes. In [31], Ordinary Differential Equation (ODE) models were proposed in order to estimate delivery delay and number of transmissions for RLC schemes and non-coding schemes, both for a single group of packets.

This paper presents new contributions that improve our understanding of the benefits of network coding in DTNs unicast application both theoretically and practically. Our main findings are summarized below:

- Leveraging event-driven graph model for DTNs ([14]), and existing results on static graphs ([16], [25]), we propose an algorithm to calculate the minimum time to deliver a group of packets, and prove a lower bound on the probability that RLC schemes achieve the minimum delivery time.
- We show that under only bandwidth constraints, the RLC scheme improves group delivery delay, but fares worse in terms of in-order packet delay and average packet delay (and in general, time to deliver a fraction of packets) and generates more transmissions in the network. At the same time, RLC schemes with replication control improve the fundamental trade-off between delay and number of transmissions.
- We study how resource constraints, and various routing design options affect the benefit of RLC schemes. In particular, RLC provides more significant benefits under substantial buffer and bandwidth constraints, limited control signaling, highly dynamic networks, and when it is applied to packets belonging to the same flow.
- The same results hold when RLC schemes are applied to multiple continuous unicast flows, provided replication control mechanisms are carefully tuned.

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

Due to space constraints, the complete description of our algorithm to calculate the minimum group delivery time, the proof of Proposition 3.2 and some details about the simulation experiments are found in the companion technical report [51].

II. Background

In this section, we first present the network model studied in this paper. We then describe the general approach to unicast routing in a DTN with and without RLC, and define the performance metrics we use to study RLC benefits. Last, we provide a discussion about the design space for DTN routing schemes. Table I summarizes the notation and the default settings used in simulation experiments.

notation	meaning	simulation setting
N	number of nodes in the network	101
$\mathcal{V}$	the set of nodes	N/A
$\mathcal{L}$	DTN contact trace	pair-wise Poisson
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	$q = 2^8$
H_{total}, η	hop count, and #. of routing decisions along minimal delay paths	N/A
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
Notations and Simulation Settings

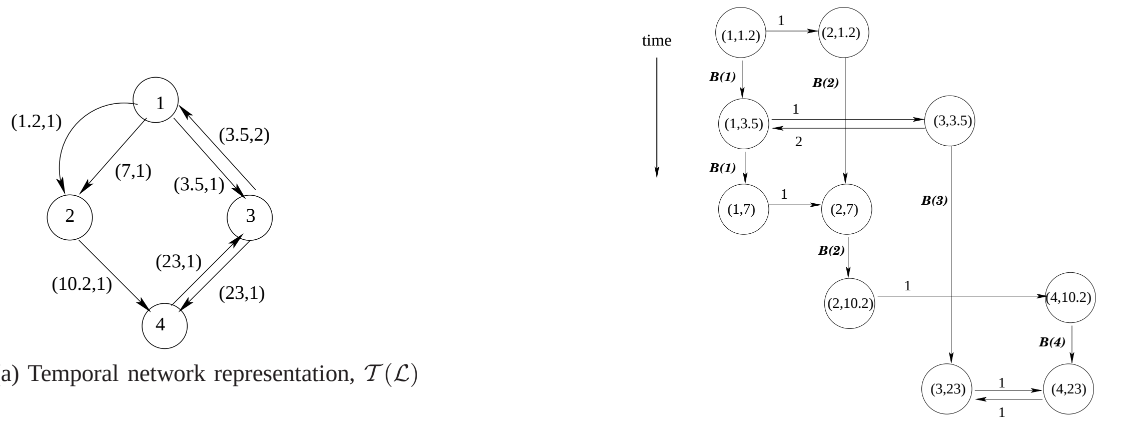
A. Network Model and Traffic Setting

We focus on unicast applications where each packet (generated by its source node) is destined to a single destination node. The network consists 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, 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 contacts, 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, as *a DTN contact trace*, denoted as $\mathcal{L} = l_1, l_2, l_3, \ldots$. Each 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, $\mathcal{B} : \mathcal{V} \rightarrow \mathbb{N}$ where $\mathcal{B}(u)$ 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.* [24]. 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

[1] 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.

represented as an edge, labeled with a pair, $(t(l), b(l))$, i.e., 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* proposed in [14]. For 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 to the graph $\mathcal{G}$, respectively denoting the sending and receiving event of the contact. A directed *inter-node edge* (depicted as a horizontal line 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 represents both temporal constraints of the contacts, and resource (bandwidth and buffer) constraints. [14] showed that many problems on DTN routing can be solved by applying classic graph theory algorithms on this static graph. The following proposition is a restatement of Theorem 4 in [14]:

Proposition 2.1: There is a feasible routing schedule for delivering K packets originated from u immediately 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 equals the number of packets sent during the corresponding contact whereas the value of a flow on an intra-node edge corresponds to the number of packets being carried by the node during the corresponding time interval.

In our simulation studies, we assume homogeneous resource constraints, i.e., $\mathcal{B}(u) = B$, for all $u \in \mathcal{V}$, and when two nodes encounter each other, $b (b \geq 1)$ packets can be exchanged in each direction. Most of our simulation results are obtained under the assumption that pair-wise meetings are described by independent Poisson processes with rate $\beta = 0.0049$[2]. The Poisson assumption 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 [12]. Later works ([22], [7]) 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 [7], [8]. Because of its tractability, the Poisson meeting process has been widely adopted ([12], [30], [40]).

B. 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. Such schemes place minimal demand on the node buffer space, and usually incur a low transmission overhead. But when future contacts 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.

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

[2] We have performed simulations using different values for β, and confirmed that the relative performance gain achieved by RLC schemes is not affect by the value of β.

minimum delivery delay when the network is lightly loaded, but causes severe resource contention under heavier traffic.

C. RLC based Routing Schemes

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

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

A collection of packets that can be linearly coded together is called a *generation*. Suppose K packets $P_i, i = 1, 2, ..., K$, constitute a generation, we denote by $\mathbf{m}_i \in \mathbb{F}_q^d$, the symbol vector corresponding to each packet. 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, $\mathbf{m}_i, i = 1, 2, ...K$, is 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 the payload is $d = \lceil S/\log_2(q) \rceil$ symbols. This leads to a relative overhead, i.e., the ratio of the size of the encoding coefficients and the payload, of $K/(\lceil S/\log_2(q) \rceil) \approx K \log_2(q)/S$.

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 the $K \times d$ matrix of the K original packets, and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_r)^{\mathrm{T}}$ is the $r \times d$ 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[3].

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 of the encoded packets 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 $\mathbf{x}_{new}$ 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 [9])[4].

D. Performance Metrics

We assume that each message generated by the application is segmented into a group of packets in order to take advantage of short contacts ([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 $i = 1, 2, ..., K$. If we assume that the destination can only process the message after all packets in the message are delivered, then an important metric is the *group delivery delay*, D_g, defined as the time from the generation of the message, i.e. of 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 might be more meaningful. For example, if the application can process each packet individually upon its delivery, then *average packet delivery delay*—$D_a = \sum_{i=1}^{K} D_i/K$—should be considered[5]. Finally, if packets must be processed by the destination in order, i.e., if P_i can be processed only after all packets $P_j, 1 \leq j \leq i$ have been delivered, then the *average in-order packet delivery delay* should be considered, where the in-order packet delivery delay for packet P_i is defined as $D'_i = \max_{1 \leq j \leq i} D_i$, for $i = 1, 2, ..., K$.

For applications that generate small messages, segmenting the message into even smaller packets can 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.

In our study, we have assumed that all information transmitted has to be delivered, and therefore delivery delay is the most important performance metric. There are network scenarios and applications where packet losses may be tolerated or have to be tolerated, so that a more relevant performance metric may be the percentage of packet delivered by a given deadline. We discuss these cases in Sec. VI.

As a measure of resources consumed (bandwidth and transmission power) 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, which is further studied in Sec. III-C.

E. Design Space

We now discuss various design options for DTN routing schemes, all of which, but for generation management, are

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

[4] If node u knows the encoding matrix of node v through full signaling (Sec. II-E), it can generate a useful combination using the deterministic algorithm proposed in [19]. We do not consider such possibility in this paper.

[5] We implement the following extension to the basic RLC operation to improve its performance in terms of packet delivery delay: if a relay node can decode one or multiple packets (before it reaches full rank), it forwards the decoded packet(s) (rather than random linear combinations) to the destination.

applicable to both non-coding and RLC schemes[6].

Control Signaling. Nodes in the DTNs periodically broadcast and listen for beacon messages in order to discover their neighbors, and exchange information about the packets/encoded-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*: signaling is limited to periodic beacon messages in order to discover neighbors. A node only transmits packets when it detects at least one neighbor.
- *Full Signaling*: after two nodes discover each other via beaconing, they exchange information about what they carry. Under the non-coding scheme, they exchange the sequence numbers of the packets they carry. The node then transmits only packets that the other does not carry. Under the RLC scheme, the nodes exchange the encoding vectors, so that each node only transmits if it has useful information for the other node.

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

Transmission Scheduling and Buffer Management. Routing schemes for resource constrained DTNs need to deal with resource contention through transmission scheduling and buffer management ([4], [26]). 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. Existing works ([4], [26], [27]) have proposed to estimate the utility of each packet, and select the packets to transmit or drop based on packet utility in order to optimize some system performance metric. These schemes typically require nodes to estimate and exchange additional control information about node mobility or packet propagation status for packet utility calculation.

In our study of the benefit of RLC, we adopt the following simple transmission scheduler for the non-coding scheme and the RLC scheme[7]. 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[8]. 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 RLC operation, i.e., a node transmits a random linear combination of the encoded-packets it carries to the other node.

As for buffer management, we consider the *drophead scheme* for non-coding schemes: when the 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 (ties are broken randomly). If the newly received packet belongs to the selected generation, one existing encoded packet of the generation is replaced by its random linear combination with the newly received packet. Otherwise, the node randomly selects two encoded packets from the chosen generation, and replaces them with their random linear combination.

Recovery Scheme. Multi-copy DTN routing schemes often employ *recovery schemes* to save resources [13], [50]. For example, under the *VACCINE* recovery, an *anti-packet* (delivery acknowledgment information) is generated by the destination upon packet delivery, and 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 provides the most significant resource savings among the different recovery schemes. We extend VACCINE recovery to work with RLC so that when a generation of packets is first delivered to its destination, the destination generates an *anti-generation* which is then propagated in the network to delete the 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. Replication control mechanisms trade-off delivery delay for resource consumption. Some of these mechanisms limit the number of transmissions by setting a maximum hop count, or a TTL timer for packet copies, while others, such as spray-and-wait, directly limit the number of transmissions.

Under binary spray-and-wait ([42], [40]), the source node assigns a number of *tokens*, denoted as C, to each source packet it generates, which specifies the maximum number of transmissions 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 copied 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 $c \leq 1$ can only deliver the packet to the destination. In this way, the total number of transmissions 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 Sec. III-C, we extend the binary spray-and-wait to be used in conjunction with RLC.

[6] For example, a network might deploy a RLC scheme that employs normal signaling, utility based transmission scheduling, VACCINE recovery, and K-hop replication control. As these design options affect routing performance and overhead, in our comparison of RLC schemes and non-coding schemes, we adopt similar design options for both of them.

[7] Any utility-based scheme can be adapted to work with RLC schemes so that transmission scheduling and buffer management decision are based on the utilities of *generations*, instead of individual packet.

[8] As verified by simulation, 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.

Generation Management. An RLC scheme needs to address the question of how many and which packets form a generation. Packets cannot be arbitrarily coded together. First, as we have observed, the overhead of transmitting and storing encoding coefficients grows with the generation size, as 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 Sec. IV-A.

III. Single Source Case

In this section, we focus on 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* for a given a contact trace under buffer constraints, 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 (Sec. III-A). We then discuss other performance metrics (Sec. III-B), and demonstrate that RLC schemes improve the *delay-per-transmission* in comparison to non-coding schemes when replication control is employed (Sec. III-C). Finally, we discuss how bandwidth and buffer constraints, different control signaling, realistic mobility traces and node churn affect the benefits of RLC schemes (Sec. III-D).

A. Probability to Achieve Minimum Group 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 a 4-tuple (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 of the K packets can be delivered to the destination. This time is in general achievable only by an oracle scheme with knowledge of all future contacts, and provides a lower bound for the group delivery time achieved by any practical routing scheme. 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 given contact trace $\mathcal{L}$ and under 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 are 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 capacities of all intra-node edges for the source node s and the destination node d to K, as we assume nodes have sufficient buffer space 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 the group of packets $(1, 4, 0, 2)$ for the DTN trace depicted in Fig. 1, with $\mathcal{B}(u) = 2, \forall u \in \mathcal{V}$. Based on Proposition 2.1, group of packets (s, d, t_0, K) can be delivered given 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:

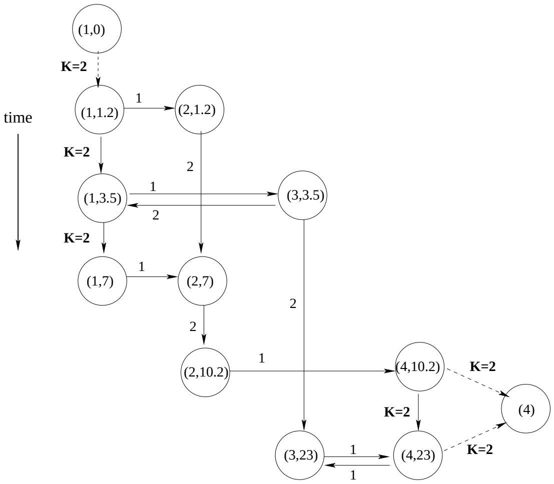


Fig. 2. Augmented event-driven graph $\mathcal{G}'(\mathcal{L}, \mathcal{B}, (1, 4, 0, 2))$ for calculating minimum group delivery time for $(1, 4, 0, 2)$, with $\mathcal{B}(u) = 2, u \in \mathcal{V}$. The newly added edges are drawn with dashed lines, and the updated intra-node edge capacity is highlighted 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 $(1, 0), (1, 1.2), (1, 3.5), (3, 3.5), (3, 23), (4, 23), (4)$.

Proposition 3.1: To determine the minimum group delivery time for the 4-tuple (s, d, t_0, K) given contact trace $\mathcal{L}$ and buffer constraints $\mathcal{B}(\cdot)$, it suffices to find $\mathcal{L}_{\min}$, the shortest left subsequence of $\mathcal{L}$ 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 group 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 ([25]). A complete description is provided in [51]. 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 events from L in time order, until FIND_PATH $(\mathcal{G}_f, (s, t_0), (d))$ finds a new path with a non-zero residual capacity[9] 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 described in Sec. II-A.

Once a path is found, the algorithm enters the *find max-flow phase* where the flow is augmented until the max-flow

[9] 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 edge. The residual capacity of a path is defined as the minimum of the residual capacities of all edges in the path.

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

```
Input: $\mathcal{L}, \mathcal{B}, s, d, t_0, K$
$\mathcal{L}_r = \mathcal{L}$, $f = 0$, $\mathcal{G}_f = \langle\{(s, t_0), (d)\}, \emptyset\rangle$
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
```

from node (s, t_0) to (d) in $\mathcal{G}_f$ is determined. While the Ford-Fulkerson algorithm [25] used here is not the most efficient max-flow algorithm, it allows us to incrementally augment the flow instead of starting 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 the Ford-Fulkerson algorithm: augmenting the flow along path P and updating the residual graph. The return value b is the increment of the flow value due to path P.

If a flow of value at least K is determined, Alg.1 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, and f yields 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 $O(K|\mathcal{L}'|)$.

Alg.1 can be 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 group delivery time. For example, the 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 support a flow of value 2 from $(1,0)$ to node (4), and correspond to a set of two paths that achieves the minimum group delivery time for the group of packets $(1,4,0,2)$.

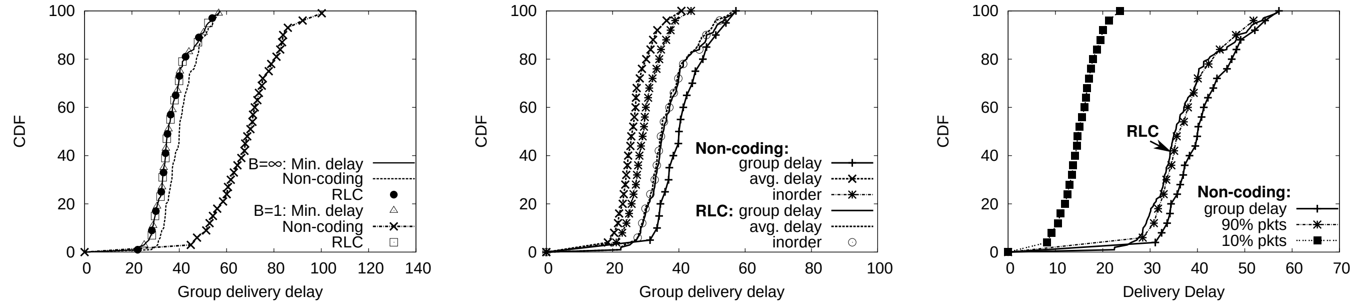
2) Probability to achieve minimum group delivery time: In practical settings, network nodes, without prior knowledge about contacts in the network, might choose "wrong" packet(s) (or encoded packet(s) for RLC schemes) to forward during a contact or to delete when the buffer is full. As a result, the destination may receive redundant information through the K *forwarding paths that achieve the minimum group delivery time*, and more time is needed to deliver the group of packets. Compared 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 is linearly dependent from the r encoded packets carried by this relay node, the probability that the 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).

In the absence of replication control, the RLC scheme makes use of all contacts to propagate the generation, including those contacts along *the set of* K *forwarding paths that achieves the minimum group delivery time*. We denote by η the number of transmission scheduling and buffer management decisions that network nodes make under the RLC scheme along this set of forwarding paths. This number affects the probability that the RLC scheme achieves the minimum group delivery time. Obviously, η is upper bounded by the total hop count (including intra-node and inter-node edges) of the set of paths, denoted as H_{total}. The actual value of η depends on the specific contact trace, buffer constraints and packet group. For the example in Fig. 2, we have $\eta = 3$ as network nodes need to make three transmission scheduling decisions, respectively during the contacts $(1,2,1.2)$, $(1,3,3.5)$ and $(2,4,10.2)$[10], and zero buffer management decision. The total number of hops of this set of paths is $H_{total} = 9 > 3 = \eta$.

We note that the RLC DTN routing scheme corresponds to a RLC routing scheme on the corresponding event-driven graph, which is a static graph. Theorem 3 in [16] applies to RLC on static graphs and provides a lower bound on the probability (in terms of finite field size, number of edges with random coefficients, and number of receivers) for an RLC scheme to support a set of feasible multicast flows. Using this result we prove the following proposition (see [51] for details):

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

[10] When node 1 encounters node 2 at $t = 1.2$, it has two packets in its buffer, and needs to decide what to transmit, therefore the contact $(1,2,1.2)$ involves transmission scheduling. Similarly for contacts $(1,3,3.5)$ and $(2,4,10.2)$. Transmission during contact $(3,4,23)$ does not involve scheduling decision as node 3 has only one packet in its buffer at time $t = 23$.



(a) Comparison of group delivery delay with and without buffer constraint

(b) Comparison of different delay metrics without buffer constraint

(c) Delay to deliver 10%, 90% packets without buffer constraint

Fig. 3. CDF of different delay metrics obtained from 100 simulation runs for a DTN with $N = 101$ nodes, $K = 10$, homogeneous exponential inter-contact time with rate $\beta = 0.0049$, bandwidth constraint of $b = 1$ packet per contact, 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 group delivery delay, and of the group delivery delay achieved by the RLC and the non-coding scheme over 100 different simulation runs for the cases with and without buffer constraints ($B = \infty, B = 1$). All simulation experiments in this paper have been carried as follows: 1) unless otherwise specified, the default parameter settings in Table I have been used, 2) at each simulation run the random number generator (used for generating the contact trace and RLC random coefficients, and for random transmission scheduling and buffer management) is initialized with a different seed, 3) the same set of contact traces is considered for all the schemes. Fig. 3(a) shows that the CDFs of the RLC scheme and of the minimum group delivery delay for both cases almost overlap. A closer inspection shows that the RLC scheme achieves the minimum group delivery delay in 98 out of 100 runs for the $B = \infty$ case and in 95 out of 100 runs for the $B = 1$ case. In contrast, the delivery delay under the non-coding scheme is larger, especially when the buffer is limited.

B. Other Performance Metrics

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, using the same set of simulations as presented in the last section, focusing on the case that buffer size are infinite.

We first consider the average packet delay and average in-order packet delay. Fig. 3(b) plots the CDFs of different delay metrics achieved by the RLC scheme and the non-coding scheme from the 100 different simulation runs (with $B = \infty$). There are four almost overlapping curves, corresponding to the 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 only slightly smaller than the group delivery delay. In this setting, the RLC scheme performs better in terms of group delivery delays, but fares worse in terms of average packet delays and in-order delivery delays. However, the RLC scheme performs better in terms of all the 3 metrics when $B = 1$ (the figure is omitted due to space constraint).

We now briefly consider the delay to deliver a certain fraction of the packets. Fig. 3(c) plots the CDFs of the group delivery delay, and of the time for the destination to receive 10% and 90% of the packets. For the RLC scheme, the 3 curves are almost identical, so that we plot a single curve. In this particular setting, the destination may receive the first packet later under the RLC scheme than under the non-coding scheme, but it can decode the first 9 packets faster. Therefore if the application requires a target delivery probability higher than 90%, the RLC scheme outperforms the non-coding scheme. We discuss this issue further in Section VI.

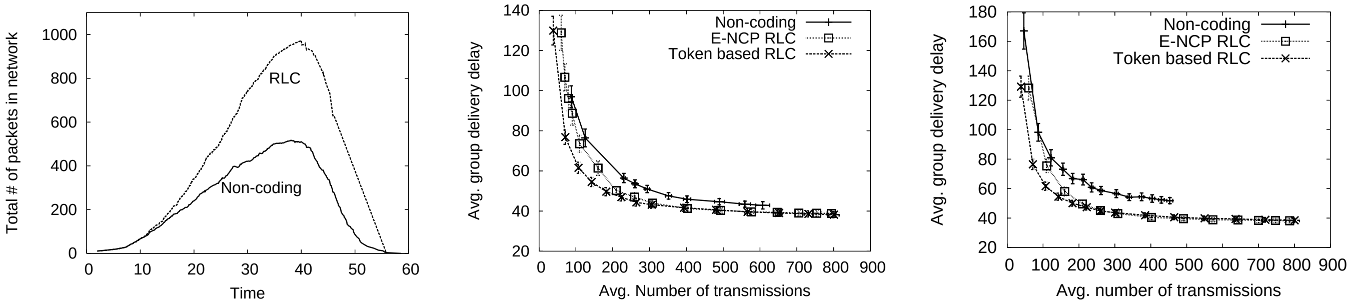
RLC schemes achieve faster information propagation at the price of a greater number of transmissions and a larger buffer occupancy. For example, Fig. 4(a) plots the total numbers of packet copies (for the non-coding scheme) or combinations (for the RLC scheme) in the network as a function of time for one simulation run (the group of packet is generated at time $t = 0$). Under the RLC scheme, the probability that two nodes that meet each other have useful information to exchange is higher, leading to a sharper increase in the total number of copies/combinations in the network. Furthermore, under the RLC scheme, 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.

C. Delay vs. Number of Transmissions Trade-off

For the RLC scheme to be beneficial in a resource constrained DTN, the RLC scheme needs to reduce delays without incurring higher transmissions overhead than the non-coding scheme.

We propose the *token-based* RLC scheme which extends binary spray-and-wait. A certain number of tokens (denoted as C_g) is assigned to each generation to limit the total number of encoded-packets that can be transmitted for the generation in the network. The operation of RLC schemes is extended with the following consideration on tokens[11]. When two *non-destination* nodes meet, they redistribute their tokens in proportion to their ranks (see [51] 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

[11] 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.



(a) Buffer occupancy, one simulation run without buffer constraint

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

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

Fig. 4. (a) Buffer occupancy under one simulation run, (b-c) Group delivery delay vs number of transmissions trade-off achieved under replication control with per-packet token number between 5 and 100 for non-coding and E-NCP schemes, and per-generation token number between 50 and 1000 for token-based RLC scheme, $K = 10$

one token. After each transmission, the sending node reduces 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, independent of the number of tokens it has. 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 $(1-1/q)^{K-1}$ (Sec. II-C). In summary, this scheme limits the total number of transmissions in the network to $C_g + K$ with high probability. The actual number is smaller when a recovery scheme is employed.

A different replication control scheme, called *E-NCP*, was proposed in [30]. For 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 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 and linearly combined at relay nodes, as under regular RLC scheme.

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 by varying the number of tokens. Fig. 4.(b) and (c) plots the average group delivery delay versus the average number of transmissions (together with the 95% confidence intervals from 100 simulation runs) achieved by the RLC scheme and non-coding scheme, respectively for the cases where there are no buffer constraints and where $B = 2$. We observe that, with a similar number of transmissions, the two RLC schemes achieve a smaller average group delivery delay than the non-coding scheme. Token-based RLC scheme outperforms E-NCP, especially under small number of transmissions. Under limited relay buffer, the RLC schemes improve the trade-off between group delivery delay and number of transmissions significantly.

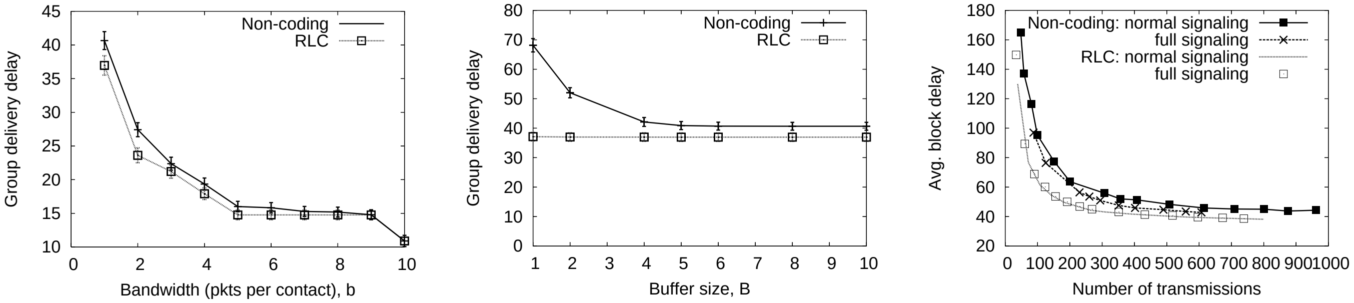
D. Discussion of RLC benefits

In this section, we study how resource constraints, signaling level, mobility patterns, and the fraction of on-and-off nodes affect the benefits of RLC schemes.

1) Impact of Different Bandwidth and Buffer Constraints: We first vary bandwidth while fixing the buffer constraint $B = K$ (i.e., no buffer constraint) and consider its impact on RLC benefit. We observe that as the network bandwidth becomes 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 ([50]). For example, Fig. 5(a) plots the average group delivery delay and its 95% confidence interval (based on 100 different simulation runs) under varying bandwidth constraints, for a group of $K = 10$ packets from the same unicast flow.

We have observed a much more significant RLC benefit when the buffer imposes a constraint (Fig. 3(a)), and hence we consider the benefit of RLC schemes as a function of buffer size. 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 as a function of node buffer size, B. We observe that as buffer sizes decrease, 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 evicts copies of other packets when buffer is full. 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. Hence, it takes much longer to deliver the "slowest" packet and therefore the whole group of packets.

2) Impact of Control Signaling: Simulation results presented so far are for the *full signaling* case, where two encountering nodes exchange information about what they carry, and decide whether and what to transmit to the other node based on such information. Full signaling incurs greater transmission and computational overheads for the RLC scheme than for the non-coding scheme, as each node needs to exchange the



(a) Group delivery delay under varying bandwidth, b, $B = \infty$

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

(c) Delay vs transmissions trade-off under full and normal signaling, $b = 1, B = \infty$

Fig. 5. Impact of bandwidth, buffer constraint and control signaling

encoding matrix (in comparison to packet sequence numbers), and calculate whether it has useful information for the other node.

We now consider *normal signaling*, where two nodes encountering each other do not exchange information about what 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 random linear combination to the other node. Fig. 5(c) plots the group delivery delay versus the number of transmissions trade-offs achieved by the non-coding and the RLC scheme with full signaling and normal signaling under varying token numbers. We observe that the non-coding scheme performs significantly worse under normal signaling, whereas the performance of the RLC scheme is almost not affected by the lack of information.

3) Impact of Real Mobility: 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 [6] testbed in the spring semester of year 2006. The DieselNet contact traces correspond to a challenging scenario where most packets cannot be delivered during a time horizon of 12 hours. The RLC scheme increases the probability to deliver a group of packets from the 24% achieved by the non-coding scheme to 31%. Our experiments are described in [51] in details.

4) Impact of Node Churn: We now briefly consider a more dynamic setting where some nodes alternate between an On state where they participate in routing and an Off state where they turn off their radios, but keep their buffered packets. We call such nodes *On-Off nodes*. The other nodes (including the source and the destination) are always active.

Fig. 6 plots the average group delivery delay under different number of On-Off nodes. For On-Off nodes, the duration of the On periods and of the Off periods is uniformly distributed respectively in $[0, 50]$ and in $[0, 100]$. When the fraction of On-Off nodes increases, the relative benefit of RLC becomes more significant both in absolute and relative values. The increased randomness of the RLC scheme make it more robust to (temporary) loss of information due to nodes being turned off.

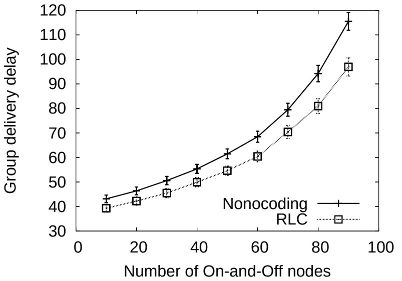


Fig. 6. Group delivery delay under varying number of On-Off nodes

IV. Multiple Unicast Flows

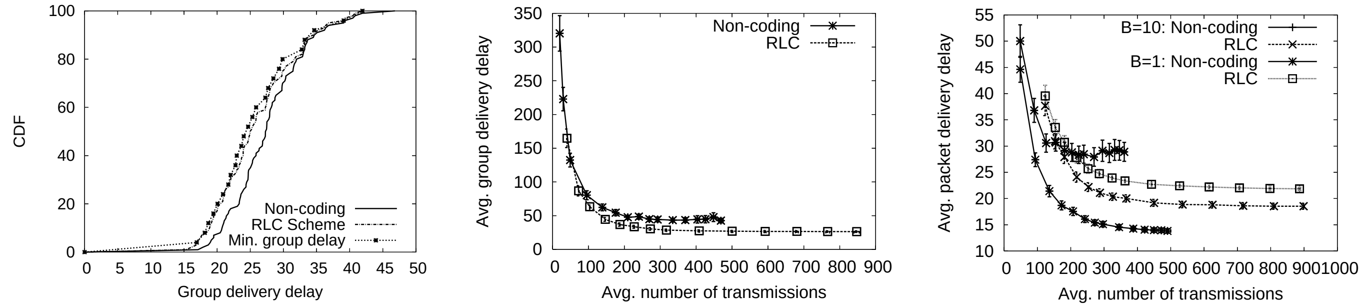
We have shown that RLC schemes achieve faster delivery of a group of packets from the same unicast flow than non-coding schemes, at the cost of a larger number of network transmissions. Furthermore, when replication control is employed, RLC schemes improve the trade-off between delivery delay and transmission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

A. Inter-Flow Coding

The focus of Sec. III is on 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., the Multiple-Sources Single-Destination (*MS_SD*) case, and



(a) CDF of group delay for MS_SD from 100 simulation runs (no replication control)

(b) MS_SD: Group delivery delay vs. number of transmissions trade-off, $B = 1$

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

Fig. 7. Benefit of RLC under inter-flow coding, for a DTN with $N = 101$ nodes, $K = 10$, homogeneous exponential inter-contact time with rate $\beta = 0.0049$, bandwidth constraint of $b = 1$ packet per contact. The per-packet token number in b) and c) is varied between $5, 10, 20, ..., 90, 100$

ii) a group of K packets originating from K different sources and destined for K different destinations, i.e., the Multiple-Sources Multiple-Destinations (*MS_MD*) case.

For the *MS_SD* case, Alg. 1 can be extended to calculate the minimum group delivery time (see [51] for details). We perform simulations 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 and of the group delivery delay 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.

The token scheme described above can be applied also to *MS_SD* and *MS_MD* cases by assigning a per-packet token number C to each of the K packets at its respective source upon packet generation. The subsequent operations are the same as 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 in the network is bounded by $CK + K$ under the *MS_SD* case, and by $CK + K^2$ under the *MS_MD* case with high probability.

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 buffers are 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*[12] versus the total number of transmissions achieved by the non-coding and the RLC scheme. Fig. 7(c) plots the results for the cases *i)* when only bandwidth is constrained ($b = 1$), and *ii)* when both bandwidth and buffer are constrained ($b = 1, B = 1$). We observe that the RLC scheme performs worse than the non-coding scheme in 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.

Given that RLC is most advantageous when applied to packets from the same flow, we focus on intra-flow coding in the case of multiple continuous flows in the next section.

B. Multiple Continuous Flows with Intra-flow coding

We now 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.

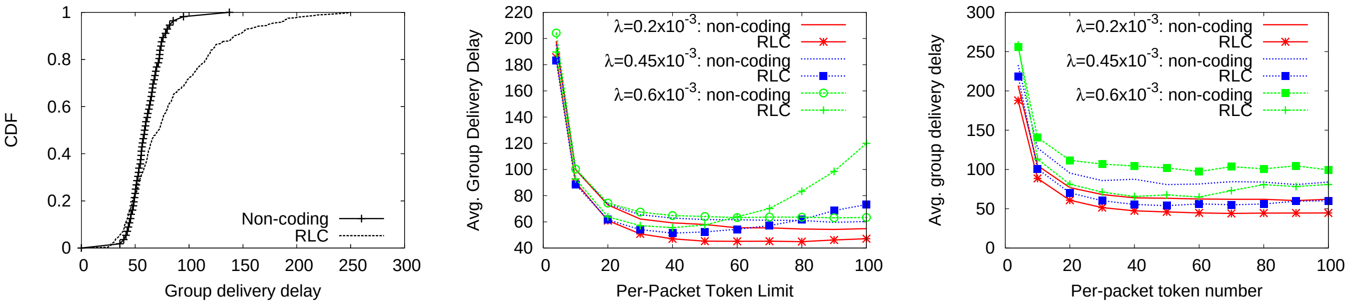
We perform simulation studies for a network with $N = 101$ nodes, assuming bandwidth constraint of $b = 1$ and no buffer constraint, to compare the average delivery delays achieved by the RLC scheme and the non-coding scheme (without replication control) under varying traffic rate λ. We observe that the RLC scheme without replication control reduces average group delivery delay when the traffic rate is low; but performs worse than the non-coding scheme when the traffic rate is high, as shown in Fig. 8(a), which plots the CDFs of group delivery delay (for all groups in the network) in steady state under the RLC scheme and the non-coding scheme for $\lambda = 0.45 \times 10^{-3}$, a relatively high 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. it is therefore more likely that under the non-coding scheme, two nodes can exchange useful information when they meet. This means that the RLC scheme achieves a smaller relative benefit. Secondly, RLC schemes incur a larger number of transmissions for each generation, and 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[13].

To alleviate resource contention, we resort to replication control. For both the RLC scheme and the non-coding scheme, we vary the per-packet token number, C, between 20 and

[12] For the MS_MD case, 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 it (i.e., *average packet delivery delay*), than the time to deliver the last packet in the group (i.e., the *group delivery delay*).

[13] 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, $\lambda = 0.45 \times 10^{-3}$, $B = \infty$

(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 ($N = 101$ flows with Poisson arrival of groups of $K = 10$ packets, $\beta = 0.0049, b = 1$

100. Fig. 8(b) plots the average group delivery delay under different per-packet token numbers for three different rates $\lambda = 0.2 \times 10^{-3}$, $\lambda = 0.45 \times 10^{-3}$, and $\lambda = 0.6 \times 10^{-3}$. Under relatively high traffic rates, the RLC scheme achieves a smaller average group delivery delay but only when the token number is carefully tuned. For example, when $\lambda = 0.45 \times 10^{-3}$, the optimal number of tokens lies between 40 and 50: if the number is too large, severe contention leads to degraded performance; if it is too small, some contacts are not exploited because all tokens have been consumed. For the non-coding scheme, when the number of tokens is smaller than 100, contention is not significant and a smaller token number leads to a larger average group delay. We do observe that when the traffic rate is high, the non-coding scheme also benefits from replication control. How to configure replication control schemes for a given network setting is an open problem ([52]), and beyond the scope of this paper.

As with the single generation case, the RLC benefit in the presence of multiple flows is more significant when the buffer is also constrained. We repeat the simulation as shown in Fig. 8(b), introducing buffer constraint of $B = 3$. The result as plotted in 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 20% for token values ranging from 20 to 100.

V. Related Work

Several works ([20], [44]) have applied erasure coding ([38], [33]) to DTNs, where the source encodes a message into a large number of blocks, such that as long as a minimum fraction of the coded blocks is received, the message can be decoded. For DTNs where there is prior knowledge about paths and their loss behavior, Jain *et al.* ([20]) 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 finding that under normal signaling, the relative benefit of RLC is much more significant than that under full signaling is in line with similar findings for broadcast applications in [45].

Using the connection between E-NCP and the low-density distributed erasure codes ([2]), [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 in E-NCP to $\Theta(\log K)$. In contrast, we compare different replication control schemes in terms of the fundamental performance trade-off between delivery delay and number of transmissions.

Lin *et al.* ([31]) developed ODE models to analyze the group delivery delay for a single group of SS_SD packets under RLC and non-coding schemes. We note that due to simplifying assumptions made in the model derivation, the models not only underestimate the delivery delays under both schemes, but also underestimate the performance difference between them.

The benefit of RLC observed in this paper is similar in spirit to that of rumor mongering ([9], [5]). For a network under 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, [9], [5] derived asymptotic bounds for the time to disseminate multiple messages under both RLC and non-coding schemes.

Finally, [29] presented a preliminary investigation on the effect of topology on the RLC performance. Simulation results 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. 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, for the case of a single group of packets (SS_SD) propagating in the network, RLC reduces the group delivery delay in comparison to non-coding schemes. In particular it achieves the minimum group delay with probability greater than or equal to $(1-1/q)^{\eta}$. Larger gains are achieved by RLC schemes when resources (bandwidth and buffer space) are severely constrained, when information about the content of other nodes is not available, when the network is highly dynamic 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 group delivery delay under multiple continuous unicast flows, with significant performance improvement when node buffer is constrained.

In our study, we have considered that all the information transmitted has to be delivered and that group delay to be the most important performance metric. There are network scenarios and applications where packet losses may be tolerated or have to be tolerated, so that a more relevant performance metric may be the percentage of packets delivered by a given deadline. In this case applying RLC to the whole group of packets may degrade the performance (as it is suggested by Fig. 3(c)) because RLC basically couples all the packets together, and then in most of the cases the destination either decodes all packets or no packet by the deadline. A possibility is to divide the set of packets to be transmitted into different generations (see Sec. II-E) and apply RLC to packets belonging to the same generation. For example if we need to transfer 1000 packets, but we are satisfied with receiving 900 packets, we could apply RLC to generations of 10 packets. We plan to investigate the issue of generation management further in our future research. Another open question is the consideration of heterogeneous mobility model.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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*, 17:1578–1591, 2009.

[8] 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)*, 2008.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16:495–510, 2008.

[24] 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.

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

[26] 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.

[27] K. Lee, Y. Yi, J. Jeong, H. Won, I. Rhee, and S. Chong. Max Contribution: On Optimal resource Allocation in Delay Tolerant Networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[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*,

Special Issue on Delay and Disruption Tolerant Wireless Communication Systems, 26(5):794–808, 2008.

[32] J. Liu, D. Goeckel, and D. Towsley. Bounds on the gain of network coding and broadcasting in wireless networks. In *INFOCOM*, 2007.

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

[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, 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] S.-Y. Kung Y. Wu, P. A. Chou. Information Exchange in Wireless Networks with Network Coding and Physical-Layer Broadcast. Technical Report MSR-TR-2004-78, Microsoft, August 2004.

[49] 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.

[50] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance Modeling of Epidemic Routing. *Elsevier Computer Networks*, 51/10:2859–2891, 2007.

[51] 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/.

[52] 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.

Xiaolan Zhang holds a Ph.D. in Computer Science from University of Massachusetts Amherst. She received her B.Sc. degree from the Beijing University in China, and her M.Sc. degree from University of Massachusetts Amherst. She is currently an assistant professor in the Computer and Information Science Department at Fordham University in New York, NY, USA. Her research interests include mobile wireless networks, mobility modeling and Disruption Tolerant Networks.

Giovanni Neglia is a researcher in Maestro team at INRIA since September 2008. He received a master's degree in electronic engineering and a Ph.D. degree in telecommunications from the University of Palermo, Italy, in 2001 and 2005, respectively. In 2005 he was a research scholar at the University of Massachusetts Amherst, visiting the Computer Networks Research Group. Before joining INRIA, he was a postdoc at the University of Palermo and an external scientific advisor in Maestro team. His research focuses on modeling and performance evaluation of computer networks and proposals of new mechanisms to improve their performance.

Jim Kurose received a B.A. in physics from Wesleyan University and a Ph.D. degree in computer science from Columbia University. He is currently Distinguished University Professor (and past chairman) in the Department of Computer Science at the University of Massachusetts. He has been a Visiting Scientist at IBM Research, INRIA, Institut EURECOM, the University of Paris, LIP6, and Thomson Research Labs. His research interests include network protocols and architecture, network measurement, sensor networks, multimedia communication, and modeling and performance evaluation. He has served as Editor-in-Chief of the IEEE Transactions on Communications and was the founding Editor-in-Chief of the IEEE/ACM Transactions on Networking. He has been active in the program committees for IEEE Infocom, ACM SIGCOMM, and ACM SIGMETRICS conferences for a number of years, and has served as Technical Program Co-Chair for these conferences. He is a Fellow of the IEEE and ACM.

Don Towsley holds a B.A. in Physics (1971) and a Ph.D. in Computer Science (1975) from University of Texas. He is currently a Distinguished Professor at the University of Massachusetts in the Department of Computer Science. He has held visiting positions at IBM T.J. Watson Research Center, Yorktown Heights, NY; Laboratoire MASI, Paris, France; INRIA, Sophia-Antipolis, France; AT&T Labs - Research, Florham Park, NJ; and Microsoft Research Lab, Cambridge, UK. His research interests include networks and performance evaluation.

He serves on the editorial boards of *Journal of the ACM* and *IEEE Journal on Selected Areas in Communications*, and has served as Editor-in-Chief of the *IEEE/ACM Transactions on Networking* and on numerous other editorial boards. He was Program Co-chair of the joint ACM SIGMETRICS and PERFORMANCE '92 conference and the Performance 2002 conference. He is a member of ACM and ORSA.

He has received the 2007 IEEE Koji Kobayashi Award, the 2007 ACM SIGMETRICS Achievement Award, the 1998 IEEE Communications Society William Bennett Best Paper Award, and numerous conference/workshop best paper awards. Last, he has been elected Fellow of both the ACM and IEEE.

Haixiang Wang received his B.E. degree in Telecommunication from Huazhong University of Science and Technology, P.R.China. He completed his M.S. degree in Computer Science from Fordham University, where he conducted research projects in Disruption Tolerant Networks under the guidance of Dr. Xiaolang Zhang. He is the recipient of Herbert Bomzer Award and Graduate Assistant Scholarship awarded by Fordham University.