

# Network Coding in Disruption Tolerant Networks

Xiaolan Zhang<sup>a</sup>, Giovanni Neglia<sup>b</sup>, Jim Kurose<sup>c</sup>

<sup>a</sup>*Dept. of Computer and Information Sciences, Fordham University*

<sup>b</sup>*INRIA, 2004 route des Lucioles, BP 93, FR-06902 Sophia Antipolis, France*

<sup>c</sup>*Dept. of Computer Science, University of Massachusetts, Amherst, MA 01008*

---

---

## 1. Introduction

In recent years, wireless communication technologies have been increasingly deployed in challenging environments where there is no communication infrastructure, as evidenced by the many efforts in building and deploying wireless sensor networks for wildlife tracking [22, 43], underwater sensor networks [38, 41], disaster relief team networks, networks for remote areas or for rural areas in developing countries [8, 10, 51], vehicular networks [4, 20] and Pocket Switched Networks [19]. Without infrastructure support, such networks solely rely on peer-to-peer connectivity among wireless radios to support data communication. Owing to limits of transmission power, fast node mobility, sparse node density and frequent equipment failures, many of such networks have only intermittent connectivity, and experience frequent disconnection of nodes. *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 - a node receiving a packet buffers and carries the packet as it moves, passing the packet on to new nodes that it encounters. When the destination node meets a node that carries the packet, the packet is delivered to the destination.

In addition to intermittent connectivity and dynamic topologies, routing in DTNs faces additional challenges due to the severe resource constraints: for the small mobile nodes carried by animals or humans, buffer space, transmission bandwidth and power are very limited; for mobile nodes in vehicle based networks, even though buffer space or power are usually not severely constrained, transmission bandwidth is still a scarce resource.

There has been a substantial amount of research on the benefits of network coding for wireless networks. For multicast applications in static wired or wireless network, Lun *et al.* [35] and Wu *et al.* [52] showed that for the problem of minimum-energy multicast, the use of network coding simplifies the problem (from an NP-complete problem to a linear optimization problem solvable

---

*Email addresses:* xzhang@fordham.edu (Xiaolan Zhang), giovanni.neglia@inria.fr (Giovanni Neglia), kurose@cs.umass.edu (Jim Kurose)

in polynomial time). For broadcast applications in mobile and static wireless networks, Widmer *et. al.* [49, 50] proposed a Random Linear Coding (RLC) [17, 18] based scheme for energy efficient broadcast. For unicast applications in static wireless network, several works [53, 24, 25, 34] have shown that network coding schemes can provide throughput gain by leveraging the broadcast nature of the wireless channel.

Because of the distinct characteristics of DTNs, some of the benefits of network coding identified above for general wireless networks do not hold for DTNs. First, due to the dynamically changing topology of DTNs, the static network model adopted in [35, 52] and the results obtained therein about coding benefit for multicast applications are not directly applicable to DTNs. Second, DTNs have sparse node density with each node usually having at most one neighbor at any instance of time, therefore the previously discovered benefit of network coding in increasing network throughput (by leveraging the broadcast nature of wireless transmission) is negligible for 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*; moreover the limited connectivity and bandwidth require DTN routing schemes to be *localized* (i.e., with only limited knowledge about local neighborhood) too. Network coding, in addition to its benefit in increasing throughput and saving cost, has been shown to facilitate the design of efficient distributed routing schemes [36].

Existing research on the application of network coding to DTNs has focused on applying Random Linear Coding to broadcast and unicast communication. In what follows we will use the expression *RLC scheme* to denote a DTN routing scheme that employs RLC, and use the expression *non-coding scheme* to denote a traditional routing scheme. For broadcast applications where all nodes are interested in receiving all packets, Widmer *et. al.* [49, 50] demonstrated the benefit of RLC schemes in improving energy efficiency. For unicast transmissions, [55, 57] demonstrated that RLC schemes can achieve faster propagation of a block of unicast packets. Furthermore, RLC schemes, when combined with binary *spray-and-wait* scheme to control the number of transmissions made in the network, improve the trade-off between delivery delay and overhead [55, 57, 31]. References [30, 31] proposed a modeling study of RLC schemes. Finally, for DTNs with Byzantine adversaries, [42] proposed a network coding based routing scheme that supports secure data communication.

The rest of the chapter is structured as follows. In Section 2, we present basic background on DTNs, non-coding based broadcast and unicast routing schemes, and Random Linear Coding. In Section 3, we introduce a taxonomy of DTN routing schemes by discussing the various design aspects of DTN routing schemes. Section 4 and Section 5 respectively present research contributions on the benefit of network coding for broadcast application and unicast application in DTNs. Section 6 discusses open issues about the application of network coding to DTNs. Finally, Section 7 summarizes the chapter.

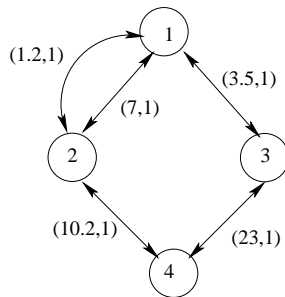


Figure 1: Temporal network representing the contacts between nodes

## 2. Background on Disruption Tolerant Networks and Random Linear Coding

In this section, we first introduce the network model, then review non-coding based routing schemes that have been proposed for broadcast and unicast communication in DTNs, and finally provide an introduction to the basic operations of RLC.

### 2.1. Network Model

Consider a network consisting of  $N + 1$  mobile nodes moving independently in a closed area according to some common mobility model such as random waypoint or random direction model [5]. 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 *meet*), they can exchange packets with each other. The *meeting time* of these two nodes is the time duration of this transmission opportunity, while the *inter-meeting time* is the duration of the time interval between two consecutive meetings, i.e., from the time instant when the two nodes go out of transmission range of each other to the time instant when they can again communicate. It has been shown in [12] that under random waypoint and random direction models, the inter-meeting time follows approximately an exponential distribution when node velocity is relatively large compared to the region size and the transmission range is relatively small. Because of the tractability of exponential inter-meeting time mobility model, it has been widely adopted (see e.g., [55, 57, 31] and references therein.).

We refer to the list of node-to-node contacts of a DTN during a certain time duration as a *DTN contact trace*, and use the so-called *temporal network* model that was originally proposed by Kempe *et al.* [26] to represent a DTN contact trace. The temporal network is a multi-graph  $G = (V, E)$  in which  $V$  denotes the nodes in the network, and  $E$  denotes the set of edges, with each edge representing a node-to-node contact. Each edge  $e \in E$  is labeled with a pair,  $(t(e), bw(e))$ , where  $t(e)$  specifies the time at which the two endpoint nodes are able to communicate, and  $bw(e)$  specifies the bandwidth constraint of the contact, i.e., the number of packets that can be exchanged over the

contact. The edges can be *directed*, if independent wireless channels are used for transmissions in the two directions, or *undirected*, if the same wireless channel is used for transmission in both directions and the total amount of capacity can be arbitrary divided between them. For example, Fig. 1 illustrates the temporal network model with directed edges for a contact trace of a DTN with four nodes during time interval  $[0, 24]$ .

Existing works study both broadcast and unicast communications in DTNs. Broadcast communication delivers each message to all nodes in the network, as in the case of content distribution service, and routing message propagation; while under unicast communication, each message is destined to a single node in the network. The investigation of network coding based DTN routing schemes has started with the simpler case where a single generation/block of (broadcast or unicast) packets is propagated in the network. With insights obtained from this simpler case, more realistic scenarios such as multiple unicast flows with continuous packet arrival process have been studied [55, 57].

## 2.2. DTN Routing Schemes Overview

Recall that DTN routing adopts a so-called “store-carry-forward” paradigm. Under this paradigm, each node in the network *stores* a packet that have been forwarded to it by another node, *carries* the packet while it moves around, and *forwards* it to other relay nodes or the destination node when they come within transmission range. In this section, we provide an overview of the basic operations of broadcast and unicast routing schemes that have been proposed in the literature, and leave the detailed discussion of various design options to Section 3.

### 2.2.1. DTN Broadcast Routing Schemes

Broadcast has long been studied in the context of wireless ad hoc networks. The early broadcast routing scheme is based on flooding, where each node in the network rebroadcasts each message it receives. In relatively dense network, this leads to excess overhead and contention [40]. It has been shown that the *gossip* based routing scheme [13], where each node rebroadcasts a message it received with a certain probability ( i.e., *forwarding probability*), reduces significantly the overhead of the routing protocol, i.e., the number of messages sent in the network.

Under a DTN, at any point of time, a node might have no neighboring nodes, therefore, the broadcast routing scheme needs to decide *when* to rebroadcast a packet depending on the control signaling adopted (Section 3). Furthermore, as each rebroadcasting only reaches one or zero neighbors, each node needs to retransmit a packet multiple times so that a certain fraction of network nodes receives the packet. For these reasons, the *probabilistic routing* proposed in [49] extended the above gossiping scheme by generalizing the forwarding probability to the *forwarding factor*,  $f$ , that can take values larger than 1. If  $f \leq 1$ , a node rebroadcasts a copy of each packet it receives once with probability  $f$ ; if  $f > 1$ , a node rebroadcasts the packet  $\lfloor f \rfloor$  times, and rebroadcasts one more

copy of the packet with probability  $f - \lfloor f \rfloor$ . Special care is taken at the source node so that a packet is transmitted at least once by the source, i.e., source node broadcasts a packet for  $\max(1, \lfloor f \rfloor)$  times, and an additional copy is transmitted with probability  $d - \lfloor f \rfloor$  if  $f > 1$ . The forwarding factor directly determines the forwarding overhead of the scheme, and clearly should be adjusted based on the node density<sup>1</sup>.

The performance metrics for evaluating different broadcast routing schemes ([49, 11]) include (i) *energy efficiency*, often measured by the total number of transmissions required to transmit a packet to all receivers, (ii) *packet delivery ratio*, the fraction of packets that are received by all the destinations, and (iii) *packet delivery delay*.

### 2.2.2. DTN Unicast Routing Schemes

For unicast communication, each packet (generated by its source node) is destined to a single node (its destination node) in the network.

Unicast routing schemes for DTNs can be classified as single-copy or multi-copy schemes. Under single-copy schemes [46], each packet is *forwarded (not copied)* along a single path, and at any point of time, there is a single copy of the packet in the network. The single-copy schemes incur a low transmission overhead and place minimal demand on the node buffer space. But when the future meeting process is 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 multi-copy schemes, a packet is *copied (i.e., duplicated)* to other nodes to be simultaneously forwarded along multiple paths to the destination; there are usually multiple simultaneous copies of a packet in the network at a given point of time. For example, the epidemic routing proposed by Vahdat and Becker [48] essentially floods the whole network in order to deliver a packet. By making use of all transmission opportunities, epidemic routing achieves minimum delivery delay when network is lightly loaded, but causes resource contention when the network is loaded. Many variations of epidemic routing that trade-off delivery delay for resource consumptions have been subsequently proposed and studied, including K-hop, probabilistic forwarding [14] and spray-and-wait [47, 44, 45] schemes.

Under multi-copy routing schemes, when a packet is first delivered to the destination, there might be multiple copies of the packet in the network. *Recovery schemes* have been proposed to delete these obsolete copies from the network to free up storage space and avoid useless transmission [14]. More details about various recovery schemes are presented in Section 3.

The performance metrics of interest for unicast applications are *packet delivery delay* and *the total number of copies or combinations made* for the packet

---

<sup>1</sup>Reference [11] proposed an adaptive scheme for each node to dynamically adjust the forwarding factor based on knowledge about its neighbors that are within two hops away.

in the network. The latter is a direct indication of the transmission overhead of a routing scheme. Clearly there exists an inherent trade-off between the packet delivery delay and the number of copies made [56].

### 2.3. Random Linear Coding

We now briefly describe the basic operation of Random Linear Coding (RLC), which is the technique considered by all existing works applying network coding to DTNs. For a more formal introduction to RLC, please refer to Chapter 1 of the book.

We assume that all packets are of the same length with  $P$  bits payload. When RLC is used in packet data networks, the payload of each packet can be viewed as a vector over a finite field [29],  $\mathbb{F}_q$  of size  $q$ , more specifically, a packet of  $P$  bits is viewed as a  $d = \lceil P/\log_2(q) \rceil$  dimensional vector over  $\mathbb{F}_q$ .

A collection of packets that may be linearly coded together by network nodes is called a *generation*. Consider a generation with  $K$  original packets,  $\mathbf{m}_i \in \mathbb{F}_q^d, i = 1, 2, \dots, K$ . 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 the addition and multiplication operations are over  $\mathbb{F}_q$ . The coefficients  $\alpha = (\alpha_1, \dots, \alpha_K)$  are called the *encoding vector*, and the resulting linear combination,  $\mathbf{x}$ , is an *encoded message*. We say that two or more encoded messages are linearly independent if their encoding vectors are linearly independent. Each original packet,  $\mathbf{m}_i$ , can be viewed as a special combination with coefficients  $\alpha_i = 1$ , and  $\alpha_j = 0, \forall j \neq i$ .

Under an RLC-based routing scheme, networks nodes store and forward encoded messages together with their coefficients. For a generation of size  $K$ , the coefficients take up  $K$  symbols; while each data packet and combination takes up  $d$  symbols, with  $d = \lceil P/\log_2(q) \rceil$ , resulting in a relative overhead (i.e., the ratio of the size of the encoding coefficients and the data packet) of  $K/(\lceil P/\log_2(q) \rceil) \approx K \log_2(q)/P$ . If in the set of encoded messages carried by a node there are at most  $r$  linearly independent encoded messages  $\mathbf{x}_1, \dots, \mathbf{x}_r$ , we say that the rank of the node is  $r$ , and refer to the  $r \times K$  matrix whose rows are the  $r$  encoding vectors as the node's *encoding matrix*,  $A$ . Essentially, this node has stored  $r$  independent linear equations with the  $K$  source packets as the unknown variables, i.e.,  $AM = X$ , where  $M = (\mathbf{m}_1, \dots, \mathbf{m}_K)^T$  is a  $K \times 1$  matrix of the  $K$  original packets, and  $X = (\mathbf{x}_1, \dots, \mathbf{x}_r)^T$  is the  $r \times 1$  matrix of the  $r$  encoded messages. When a node (e.g., the destination) reaches rank  $K$  (i.e., full rank), it can decode the original  $K$  packets through matrix inversion, i.e., solve  $AM = X$  for  $M = A^{-1}X$  using standard Gaussian elimination algorithm <sup>2</sup>.

---

<sup>2</sup> It is possible that the destination node decodes an original packet before the matrix reaches full rank, as long as the encoding matrix  $A$  contains a vector that has exactly one non-zero coefficient.

We illustrate the data forwarding using the transmission from node  $u$  to node  $v$  as example. Node  $u$  generates a random linear combination ( $x_{new}$ ) of its currently stored combinations, say  $x_1, \dots, x_r$ :  $x_{new} = \sum_{j=1}^r \beta_j x_j$ , where the coefficients  $\beta_1, \dots, \beta_r$  are chosen uniformly randomly from  $\mathbb{F}_q$ . Clearly,  $x_{new}$  is also a linear combination of the original  $K$  packets. This new combination, along with the coefficients *with respect to the original packets*, is forwarded to node  $v$ . If there is at least one combination stored in node  $u$  that cannot be linearly expressed by the combinations stored in node  $v$ , then node  $u$  has useful (i.e., *innovative*) information for the node  $v$ , and the new random combination  $x_{new}$  is useful to node  $v$  (i.e., can increase the rank of node  $v$ ) with probability greater or equaled to  $1 - 1/q$  [9]<sup>3</sup>.

The RLC scheme incurs computation overhead as nodes perform random linear combinations and the destination node performs decoding operations. While the complexity of the encoding operation grows linearly with the generation size, the decoding operation has quadratic complexity in generation size.

### 3. Design Space

Before presenting the main findings about the benefits of RLC in DTNs, we discuss the different design options for DTN routing schemes to explore the design space of DTN routing schemes. All these design options except generation management are applicable to both non-coding schemes and RLC schemes. All design options except recovery scheme are applicable to both broadcast and unicast schemes.

**Generation Management.** A coding based scheme needs to address the question of how many and which packets form a generation, i.e., generation management. Packets cannot be arbitrarily coded together for the following two reasons. First, the overhead of transmitting and storing encoding coefficients grows with the generation size, and so does the computational complexity of decoding algorithm. Second, for unicast applications, when  $K$  packets belonging to  $K$  different (unicast) flows are coded together, a destination has to receive  $K$  coded packets just to decode the one packet destined to it. The second consideration does not apply to broadcast communication, where all nodes are interested in receiving all packets.

**Control Signaling.** Because of the ad hoc nature and dynamically changing topology of DTNs, nodes perform beaconing in order to discover their neighbors (via broadcasting periodic beacon packets), and/or exchange with neighbors information about packets/coded-packets they carried. Such control signaling is useful for nodes to decide *whether to transmit and what information to transmit*.

---

<sup>3</sup> Knowing the encoding matrix of node  $v$ , node  $u$  can iteratively generate random linear combinations from its stored combinations until a combination useful to node  $v$  is generated. Alternatively, node  $u$  can also generate a combination useful to node  $v$  by using the deterministic algorithm proposed in [21]. Such processing trade-off computational overhead for savings in transmission bandwidth.



The following different levels of control signaling have been considered in the literature:

- *No Signaling*: Under this most basic case (referred to as *no beacon* in [49]), no information about the neighborhood is available. Nodes decide to transmit packets without knowing whether there is a neighboring node or not.
- *Normal Signaling*: Under normal signaling (referred to as *normal beacon* in [49]), each node periodically transmits beacon messages in order to discover neighboring nodes, i.e., nodes within its transmission range. With normal signaling, a node typically only transmits information when it detects at least one neighbor.
- *Full Signaling*: Under full signaling (referred to as *intelligent beacon* in [49]), each node not only performs periodic beaconing to discover its neighbors, but also exchanges with its neighbors information about what packets or coded packets are stored locally, i.e., the sequence numbers of packets or the encoding vectors of coded packets. Based on such information, a network node typically only transmits to its neighbors if it has useful information for them.

**Replication Control.** For 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* mechanism.

In the probabilistic routing scheme proposed in [49] for broadcast applications, the replication control is through the forwarding factor.

Spray-and-wait schemes ([47, 45, 55]) adopt a different replication control mechanism, where the total number of times a packet is transmitted in the network is directly controlled. Under the binary spray-and-wait scheme [47, 45], the source node assigns a counter value (a number of *tokens*), denoted as  $L$ , 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  $l$  ( $l > 1$ ) meets another relay node that does not carry a copy of the packet, the packet is forwarded to the latter node and the  $l$  tokens are equally split between the two copies of the packet<sup>4</sup>. A node carrying a packet with token value of 1 does not forward the packet to relay nodes; it only deliver the packet to the destination. In this way, the total number of copies made for the packet in the whole network is bounded by  $L$ , though the actual number of copies being made is often smaller than  $L$  when a recovery scheme is employed. In Section 5.2, we review and compare two different replication control schemes based on the binary spray-and-wait scheme, that have been proposed to be used in conjunction with RLC.

---

<sup>4</sup>If  $l$  is odd, the former copy keeps  $\lceil l/2 \rceil$  tokens and the new copy is assigned  $\lfloor l/2 \rfloor$  tokens.



**Transmission Scheduling and Buffer Management.** Routing schemes running on DTNs with resource constraints need to deal with resource contentions through transmission scheduling and buffer management [3, 28]. When a node encounters another node, the scheduling mechanism 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 original packet or coded packet, it decides whether and how to make space for the new packet based on the buffer management policy [15]. Different transmission scheduling and buffer management schemes result in different system performance, such as system wide average delivery delay.

In [31], the following different scheduling policies are considered for non-coding schemes:

- *random policy* that chooses each packet with the same probability,
- *local rarest policy* where each node chooses to transmit the packet that it has transmitted least up to that time instant,
- *global rarest policy* an oracle scheme where a node chooses the packet that has the smallest number of copies in the network.

Focusing on the potential benefits of network coding, [55, 57] have studied round robin scheduling for the source packets and randomized transmission scheduling for relay packets. This means that the source node takes turns to transmit each of the source packets during encounters<sup>5</sup>, and a relay node selects uniformly at random a packet or a generation to transmit during an encounter.

For the buffer management, a drophead scheme has been considered by [55, 57]: when a node with a full buffer receives a packet, it drops the relay packet<sup>6</sup> that has resided in the buffer the longest. Under an RLC scheme, when a node with a full buffer receives a combination, it randomly combines it with a combination existing in the buffer and replaces the existing combination with the new combination.

**Recovery Scheme.** For unicast application, multi-copy DTN routing scheme such as epidemic routing and spray-and-wait scheme often employs a *recovery scheme* to save resource consumption [14, 56]. For example, under the *VACCINE* recovery scheme [14, 56], an anti-packet is generated by the destination when it first receives a packet, which is then propagated in the entire network, in the same fashion that a data packet is propagated under epidemic routing, to delete obsolete copies of the packet. Among the different recovery schemes, *VACCINE* recovery leads to the most significant resource savings, and therefore is adopted by existing works on network coding’s benefit in DTN unicast application. In particular, [55] extended *VACCINE* (and any other) recovery scheme

---

<sup>5</sup>With round robin scheduling at the source, all source packets are given equal opportunity to be disseminated into the network, yielding a smaller block delivery delay than the purely randomized scheduling [57].

<sup>6</sup>It is often assumed that network nodes have sufficient storage to store its own source packets.

to work on RLC schemes: when a generation of packets is first delivered to its destination(s), acknowledgement information (called *anti-generation*) is generated by the destination(s) and propagated in the network to delete remaining copies of the packets or combinations of packets that belongs to the generation.

#### 4. Coding Benefits for Broadcast Communication

In [49, 11], the authors investigated the benefit of network coding for broadcast applications in wireless networks, considering different scenarios such as static or mobile wireless networks with different node densities. In this section, we present their results for sparse mobile wireless networks, a type of DTNs. We first review the theoretical result in [11] which show that RLC schemes require on average fewer transmissions to reach all the nodes, and then present simulation studies that examine the effects of replication control, control signaling level, and mobility models.

##### 4.1. Coding Benefits in Energy Efficiency

For broadcast applications in DTNs, network coding based schemes deliver all messages to all nodes with fewer transmissions than non-coding schemes, and therefore improve energy efficiency [11].

Consider a network of  $N$  nodes, where each node has generated a packet to be broadcast to all the other nodes. Assume nodes move according to the *uniform at random* mobility model, i.e., at each time slot each node independently jumps to a new location in the terrain selected uniformly at random. At each time slot, each node decides to turn off or on its radio respectively with probability  $p$  and  $1 - p$ . Assume there is no control signaling (no information about neighboring nodes and the information they carry). In each time slot, each node that is turned on randomly chooses a packet to transmit (under non-coding scheme), or transmits a random linear combination of its coded packets to its neighbors (under RLC scheme). There are on average  $(1 - p)N$  transmissions in the network at each time slot.

**Theorem 1.** [11] *Broadcasting to all receivers can be achieved using on the average*

- $T_w = \frac{N \log N}{(1-p)^2}$  time slots, without using network coding,
- $T_{nc} = \frac{\Theta(N)}{(1-p)^2}$  time slots, using network coding with a large enough field size,  $q$ ,

*Thus, on average, the ratio of the time slots needed to broadcast to all receivers without and with network coding is*

$$\frac{T_w}{T_{nc}} = \Theta(\log N).$$

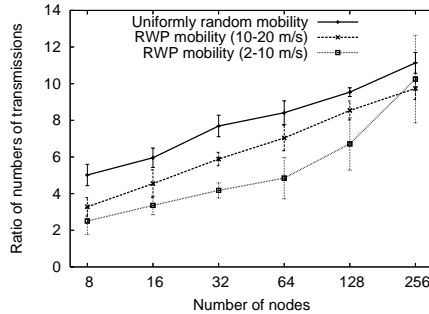


Figure 2: Ratios of transmissions required by simple flooding and an RLC broadcast scheme under uniform at random mobility model and Random Waypoint Mobility (RWP, with different average velocity) (Fig. 8 in [11])

Note that the ratio  $T_w/T_{nc}$  is equal to the ratio of the total number of transmissions required to broadcast to all the receiver in the two cases (without and with RLC), because the average number of transmissions at each time slot is same under the two schemes, i.e.,  $N(1-p)$ .

As observed in [11], the benefit of RLC in the above setting is similar in spirit to that of algebraic gossip, a RLC based protocol for message dissemination [9]. Both problems can be viewed as special instances of the *coupon collector's problem* that considers drawings with replacement from a set of  $N$  different coupons and studies the number of trials needed to obtain all  $N$  coupons. On the average, collecting all the coupons requires  $O(N \log N)$  drawings [39]. Basically, as one has drawn more and more coupons, the probability that a new coupon is drawn in the following trial becomes smaller and smaller. By contrast, RLC schemes with a large enough field correspond to a modified version of the coupon collector's problem where each new trial brings a new coupon with high probability, therefore one only needs  $O(N)$  trials in order to collect all  $N$  coupons [9].

#### 4.2. Practical RLC Broadcast Scheme

The performance comparison of simple routing schemes in the previous section considers the uniform at random mobility model, and assumes no control signaling or replication control mechanism. For more practical settings where nodes move according to random waypoint mobility model, and furthermore, when control signaling and replication control mechanism are employed, [49, 11] carried out simulation studies to evaluate the benefit of RLC schemes.

Under the uniform at random mobility model, nodes are completely reshuffled at each time slot. Common mobility models such as Random Waypoint Model (RWP) and random direction model exhibit similar memoryless property, when the transmission range is small in comparison to the region where nodes move and/or in comparison to the distance traveled by a node during a beaconing period (see Section 2.1). We expect the RLC scheme to provide

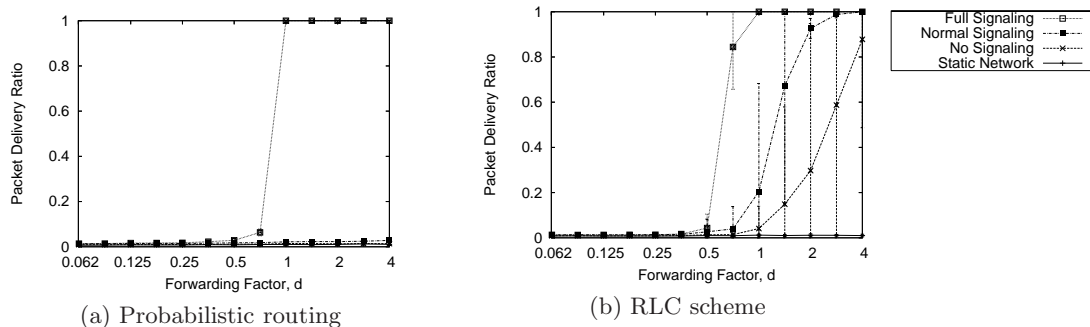


Figure 3: Packet delivery ratio under varying forwarding factor in sparse networks (Fig. 4 in [49])

similar benefit as under the uniform at random mobility model. However, when node movement is slower and the transmission range is larger, data packets are less well “mixed”, leading to less significant benefits of RLC scheme. Fig. 2 plots the ratios of the number of transmissions required by the non-coding and the RLC scheme under three different mobility settings. We observe that the relative gain of RLC scheme under RWP is smaller than that under the uniformly random mobility model. In particular, the slower the node velocity, the smaller the gain, because the nodes shuffling is reduced.

We now consider the benefit offered by RLC when control signaling and replication control are introduced. Fig. 3 plots the *packet delivery ratio* achieved by (a) the probabilistic routing (i.e., non-coding) scheme, and (b) the RLC scheme, when the amount of replication is controlled by the *forwarding factor* as introduced in Section 2.2.1. We observe that both non-coding and RLC schemes perform poorly under static setting (i.e., the “Static Network” curve). On the contrary, when nodes have a RWP mobility with zero pause time and a minimum and maximum speed of 2 m/s and 10 m/s, respectively, RLC scheme outperforms non-coding scheme for all three control signaling. The benefits of network coding are remarkable when no signaling is performed or only normal signaling is adopted. For example, if no signaling is employed, while non-coding scheme fails to deliver almost all packets with forwarding factor as large as 4, RLC scheme delivers 80% of the packets with forwarding factor of 4. When full signaling is used, both schemes achieve 100% delivery ratio with a forwarding factor of 1. This demonstrates that RLC facilitates the design of an efficient, low-complexity, distributed broadcast routing protocol. We also observe that the RLC scheme provides a smoother trade-off between forwarding overhead and packet delivery ratio.

## 5. Coding Benefits for Unicast Applications

In this section, we present research contributions on the benefits of network coding for unicast applications in DTNs. The main focus is on the simple

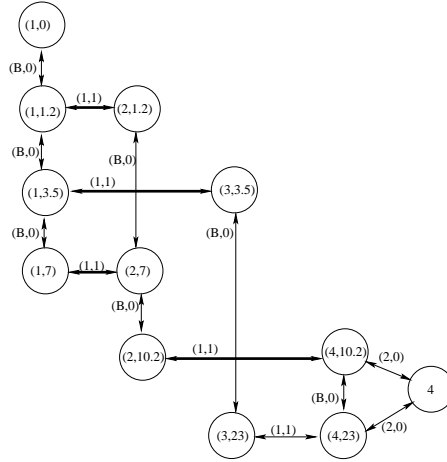


Figure 4: Event-driven graph for calculating minimum block delivery delay for block  $(1, 4, 0, 2)$  under the contact trace shown in Fig. 1, assuming each node can store at most  $B$  relay packets. Links over which random linear combinations are transmitted are drawn in thick lines (there are  $\eta = 4$  such links), e.g., over link  $(1, 1.2) \rightarrow (2, 1.2)$ , node 1 transmits a random linear combination of the two packets to node 2. Over link  $(3, 23) \rightarrow (4, 23)$ , node 3 does not perform RLC as it has only one encoded-packet in its buffer.

case where there is a single unicast flow made up by a block of  $K$  packets, propagating in a DTN where bandwidth and buffer are constrained. We use the 4-tuple  $(s, d, t_0, K)$  to denote a block of  $K$  unicast packets that is generated by source node  $s$  at time  $t_0$ , all of which destined to the same destination node  $d$ . During each contact,  $b$  ( $b < K$ ) packets can be exchanged. Note that without this limitation, the block could spread as a single message and there would be no need to perform network coding. Each node can carry  $B$  ( $B < K$ ) relay packets but has enough buffer space to store packets originating from or destined to itself. The performance metrics of interest are the *block delivery delay*, i.e., the time to deliver the block of packets, and the *total number of copies or combinations made*.

In Section 5.1 we show that applying RLC to the block of packets reduces block delivery delay. Then in Section 5.2, we demonstrate that when a replication control scheme is employed, RLC schemes improve the delivery delay versus transmission number trade-off, and achieve smaller network wide average block delivery delay in multiple generation case. We discuss how bandwidth and buffer constraints, block size, generation forming, and different control signaling levels affect the relative benefits and the overhead of RLC schemes in Section 5.3. In Section 5.4, we present modeling study that characterizes the performance of RLC scheme. Finally, Section 5.5 reviews other research works that applied network coding to DTN unicast communications.

### 5.1. Network Coding Reduces Block Delivery Delay

In this section, we demonstrate the benefits of RLC schemes in reducing block delivery delay. For a given block of packets,  $(s, d, t_0, K)$ , and a fixed contact trace, there is a *minimum block delivery delay*, achievable by a centralized oracle scheme with knowledge of all future contacts, and a lower bound for block delivery delay achieved by any routing scheme (Section 5.1.1). Thanks to the increased randomness in data forwarding, RLC schemes achieve the minimum block delivery delay with high probability (Section 5.1.2). Finally, we discuss the performance of RLC schemes in terms of other metrics in Section 5.1.3.

#### 5.1.1. Minimum Block Delivery Delay

For a block of packets specified by a tuple  $(s, d, t_0, K)$ , i.e., a block of  $K$  packets generated by a source  $s$  at time  $t_0$  and destined to a destination  $d$  under a given contact trace, the minimum block delivery delay can be calculated using an algorithm proposed in [57].

The algorithm constructs a time-independent *event-driven graph* based on the contact trace as follows [16] (Fig. 4 illustrates the event-driven graph for the contact trace in Fig. 1.). Starting with an empty event-driven graph, the contact events in the contact trace are processed according to their time order. For a contact between mobile nodes  $i$  and  $j$  at time  $t$ , two nodes labeled as  $(i, t)$  and  $(j, t)$ , and a link connecting them is inserted into the graph. The label of the link indicates the number of packets that can be transferred in each direction during the contact. Moreover, each of the two nodes  $((i, t)$  and  $(j, t))$  is connected to the node corresponding to the previous contact of the same mobile node (if any). The labels of these edges represent again the maximum number of packets that can be “transferred”, the first value corresponds to the buffer size of the node, while the second value is always equal to 0<sup>7</sup>. For example, for the contact between node 1 and 3 at time 3.5, nodes  $(1, 3.5)$  and  $(3, 3.5)$  are inserted into the graph, with a link labeled with  $(1, 1)$  connecting them.  $(1, 3.5)$  is also connected to  $(1, 1.2)$  with a link labelled as  $(B, 0)$ , given that each node can store  $B$  relay packets. In order to calculate the block delivery delay for a block of packets specified by tuple  $(s, d, t_0, K)$ , two special nodes  $(s, t_0)$  and  $d$  are added into the graph (e.g., node  $(1, 0)$  and 4 in Fig. 4).

The event-driven graph is a static graph that not only captures the temporal order of the contacts in the contact trace, but also represents the bandwidth and buffer constraints. The propagation of packets under the contact trace corresponds to the propagation of packets in this static graph: a mobile node (say node 1) transmitting a packet (coded or original) to another mobile node (say node 2) at a particular time instance ( $t = 1.2$ ) corresponds to the transmission of the packet over a horizontal link  $((1, 1.2) \rightarrow (2, 1.2))$ ; a mobile node (say node 2) buffering a packet until a future contact (at  $t = 7$ ) corresponds to the transmission of the packet over a vertical link (such as  $(2, 1.2) \rightarrow (2, 7)$ ) in the static graph. As a result, classical graph algorithms can be applied to the

---

<sup>7</sup>A packet received by node  $i$  at time  $t$  cannot be transferred to this node in the past.

graph [16] to solve various networking problems. In particular, [57] show how the event-driven graph can be advantageously used to calculate the minimum block delivery delay for  $(s, d, t_0, K)$ . First, we observe that the existence of a flow of value  $k$  from  $(s, t_0)$  to  $d$  in the event-driven graph implies the existence of a set of forwarding decisions that enable to transfer  $k$  packets from the source to the destination using the contacts that have been considered to generate the event-driven graph. In particular, if these contacts occur in the time range  $[t_0, t]$ , the existence of this flow of value  $k$  implies the possibility to transfer the  $k$  packets generated at time  $t_0$  by time  $t$ . Given these considerations, the minimum block delivery time for  $(s, d, t_0, K)$  is the earliest time  $t'$  for there to be a flow of value  $K$  from  $(s, t_0)$  to  $d$  in the event-driven graph built with the contacts in  $[t_0, t']$ . The minimum block delivery delay is the minimum block delivery time  $t'$  minus the initial time instant  $t_0$ .

The algorithm in [57] finds the minimum block delivery delay iteratively. Starting with an empty event-driven graph, the algorithm scans the contact trace from the block generation time,  $t_0$ , and gradually enlarges the graph by considering contact events according to their time order until a set of paths with a total capacity of  $K$  from the node  $(s, t_0)$  to the node  $d$  is found in the event-driven graph (using the Ford-Fulkerson algorithm [7]). Upon termination, the algorithm returns the time of the last contact considered, which is the minimum block delivery time.

### 5.1.2. Probability to Achieve Minimum Block Delivery Delay

We now consider how randomized non-coding and coding schemes perform in the practical setting where nodes have no knowledge about future contacts. We use a block of packets,  $(1, 4, 0, 2)$ , under the contact trace shown in Fig. 1 as example. The minimum block delivery delay is 23 as only after the contact at time 23 is processed that the maximum flow from node  $(1, 0)$  to node 4 in the event-driven graph (Fig. 4) reaches 2.

We first consider the non-coding scheme with random scheduling at relay nodes and round robin scheduling at source node, assuming full signaling is used. Node 1 first forwards  $m_1$  to node 2 at time  $t = 1.2$ , then forwards  $m_2$  to node 2 at time  $t = 3.5$ , and finally forwards  $m_2$  to node 2 at time  $t = 7$  (as node 2 already has  $m_1$ ). When nodes 2 and 4 meet at  $t = 10.2$ , node 2 randomly selects a packet ( $m_1$  or  $m_2$ ) from its buffer to deliver to node 4. There are two possibilities:

(i) with probability 0.5, packet  $m_1$  is selected to be delivered to node 4, and the minimum block delivery delay is achieved when node 3 delivers packet  $m_2$  at  $t = 23$ ;

(ii) with probability 0.5, packet  $m_2$  is selected. As a result, when node 3 meets node 4 at  $t = 23$ , it has no useful information for node 4. Hence, the non-coding scheme achieves the minimum delay with probability 0.5.

Under the RLC scheme, node 1 forwards random linear combinations  $c_1$  and  $c_2$  to node 2, and  $c_3$  to node 3 at times  $t = 1.2, 7$ , and  $3.5$  respectively. Node 3 transmits  $c_3$  to node 4 at time  $t = 23$ . We distinguish two cases depending on if  $c_1$  and  $c_2$  are independent or not.

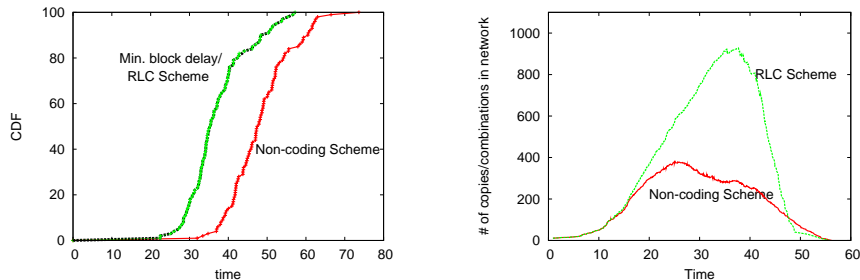


(i) If  $c_1$  and  $c_2$  are independent, node 2 stores both combinations, and generates a random linear combination  $c_{12}$  of  $c_1$  and  $c_2$  and forwards it to node 4, when it meets node 4. As  $c_3$  can be linearly expressed by  $c_1$  and  $c_2$  (given that  $c_1$  and  $c_2$  are independent),  $c_{12}$ , the random linear combination of  $c_1$  and  $c_2$ , is independent of  $c_3$  with probability  $1 - 1/q$ . We therefore conclude that for this case, with probability  $1 - 1/q$ , node 4 can decode the two original packets from  $c_{12}$  and  $c_3$  at time  $t = 23$ .

(ii) If combinations  $c_1$  and  $c_2$  are linearly dependent, then node 2 only stores  $c_1$ , which is forwarded to node 4 at  $t = 10.2$ . Then, with probability  $1 - 1/q$ ,  $c_3$  and  $c_1$  are independent, and node 4 reaches full rank at  $t = 23$ . Being that the two cases are exclusive and exhaustive, we conclude that for this contact trace, the RLC scheme achieves minimum block delivery delay for block  $(1, 4, 0, 2)$  with probability  $1 - 1/q$ . For  $q = 2^8$ , a commonly used finite field size, this probability is much larger than the probability of 0.5 achieved by the non-coding scheme.

From this example, we observe that RLC schemes provide much larger randomness than non-coding schemes, as the network nodes randomly and independently combine their stored coded packets to generate and forward coded packets, and the number of independent coded packets is much greater than the number of packets,  $K$ , from which non-coding schemes choose one to forward. As a result, the probability that a piece of redundant information is forwarded is much smaller for RLC schemes than for non-coding schemes.

In general, the probability that an RLC scheme achieves the minimum block delivery delay depends on the contact trace, more specifically, on the sequence of contacts that occur in the network after the block generation time. In order to have more insight into RLC schemes, we provide a different way to look at information propagation. Consider again the block  $(1, 4, 0, 2)$  under the contact trace shown in Fig. 1. The RLC based DTN routing scheme corresponds to a RLC based transmission scheme on the corresponding static graph shown in Fig. 4. A transmission over horizontal link such as link  $(2, 10.2) \rightarrow (4, 10.2)$ , corresponds to the transmission from a mobile node (node 2) to another mobile node (node 4) at the particular time instance ( $t = 10.2$ ). Such transmission involves RLC operations if the sender node combines its stored packets before forwarding it to the receiver node. On the other hand, a transmission over a vertical link in the static graph represents data packets being buffered at the mobile node, and therefore does not involve RLC operation. The RLC scheme in the static graph achieves the maximum flow of  $K = 2$  from node  $(1, 0)$  to node 4 if and only if the RLC scheme delivers the block of  $K = 2$  packets with the minimum block delivery delay in the DTN contact trace. The former problem is a special case of the general multiple-source multicast connection problem considered in [18]. This problem considers the transmission of a set of multicast packets originating from different source nodes to a common set of receiver nodes in a (static) graph  $G = \langle V, E \rangle$  where each link has a certain capacity limit. If a given multicast flow, specified by the set of source nodes, the set of common receiver nodes, and the data rates of the sources, can be supported by the network, the following theorem in [18] provides a lower bound



(a) CDF of block delivery delay, 100 runs    (b) Storage consumption under one run

Figure 5: DTN with  $N = 101$  nodes, homogeneous exponential inter-meeting time with rate  $\beta = 0.0049$ , bandwidth constraint of  $b = 1$  packet per contact, and unlimited buffer space.

on the probability that the RLC scheme supports the multicast flow.

**Theorem 2.** ([18]) *Consider a multicast connection problem on an arbitrary network with independent or linearly correlated sources, and a network code in which some or all network coding coefficients are chosen uniformly at random from a finite field  $\mathbb{F}_q$  where  $q > d$  ( $d$  is the number of multicast receivers), and the remaining code coefficients, if any, are fixed. If there exists a solution to the network connection problem with the same values for the fixed code coefficients, then the probability that the random network code is valid for the problem is at least  $(1 - d/q)^\eta$ , where  $\eta$  denotes the number of links associated with random coefficients.*

Note that  $\eta$  here corresponds to the number of links<sup>8</sup> along the set of source-to-destination paths (of total capacity  $K$ ) that perform RLC operation. This theorem can be applied to our setting, by setting  $d = 1$  for the unicast communication. Given that by the minimum block delivery time, the static graph can support a flow of capacity equal to the size of the block, based on the above theorem, we conclude that the probability that the RLC scheme achieves the maximum flow value is at least  $(1 - 1/q)^\eta$ . In other words, the RLC scheme achieves the minimum block delivery delay with a probability at least  $(1 - 1/q)^\eta$ . This result demonstrates the underlying connection between the benefits of RLC in two seemingly different network settings, i.e., traditional static networks and DTNs with dynamic network topologies.

Simulation results have confirmed that the block delivery delay under the RLC scheme is very close to the minimum block delivery delay, as shown in Fig. 5(a), which plots the empirical cumulative distribution function (CDF) of minimum block delivery delay, and the block delivery delay achieved by RLC and non-coding scheme over 100 different simulation runs each with a different random seed.

<sup>8</sup>A link that can transmit  $B > 1$  packets is counted as  $B$  links.

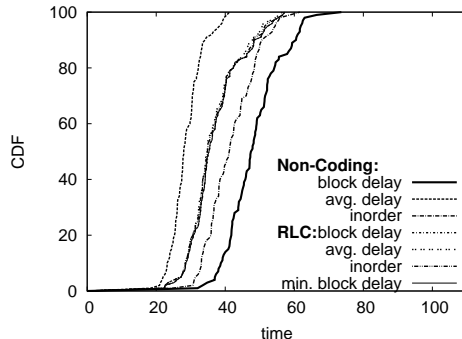


Figure 6: CDF of different delay metrics of 100 simulation runs, under the same network setting as that of Fig. 5.

### 5.1.3. Other Metrics

Having seen that the RLC scheme achieves smaller block delivery delay than the non-coding scheme, we now consider other performance metrics.

We first consider average packet delay and average in-order packet delay. Fig. 6 plots the empirical CDFs of different delay metrics achieved by the RLC scheme and the non-coding scheme over 100 different simulation runs. We observe that, under the RLC scheme, all three metrics, i.e., block delivery delay, average delay and in-order packet delay, are almost identical, and very close to the minimum block delivery delay. On the other hand, under the non-coding scheme, there is significant difference between average packet delay, in-order packet delay and block delivery delay. The RLC scheme performs better than the non-coding scheme in terms of block delivery delay and in-order packet delay (the improvement in terms of block delivery delay is larger than that of in-order packet delay), but performs worse 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 its matrix reaches full rank, it forwards the decoded packets (rather than random linear combinations of its coded packets) to the destination.

The faster information propagation of the RLC scheme is achieved at the price of more transmissions and larger buffer occupancy. For example, Fig. 5(b) plots the total number of packet copies (for the non-coding schemes) or combinations (for the RLC scheme) in the entire network as a function of time for one simulation run. Under the RLC scheme, network nodes randomly combine packets before forwarding them, as a result, when two nodes come into contact with each other, they are more likely to have useful information to exchange. This results in a sharper increase in the number of copies/combinations in the network under the RLC scheme. Furthermore, for the RLC scheme, the recovery process (VACCINE recovery is used here) starts only when the whole generation is delivered, whereas for the non-coding scheme, the recovery process for

an individual packet starts immediately when the packet is delivered.

### 5.2. Network Coding Improves Delay vs. Transmission Number Trade-off

In this section, we show that when a replication control mechanism is employed to control the number of transmissions, the RLC scheme improves the delay versus number of transmissions trade-off, i.e., the RLC scheme achieves smaller average block delivery delay than the non-coding scheme with a similar number of transmissions in the network [55, 57].

We first consider the replication control mechanism proposed in [55, 57], the so-called *token-based* RLC scheme, which is based on the binary spray-and-wait scheme. In this scheme, the source node assigns a number of tokens (denoted as  $L_g$ ) to each generation of packets that it generates, which limits the total number of combinations that can be exchanged for this generation in the network. Data forwarding follows the basic operation for RLC schemes described in Section 2.3, complemented with a specific token management policy. For the sake of simplicity, we consider a single generation, so that we can talk about the number of tokens and the rank of a node without the need to specify the generation. Consider, for example, a meeting between node  $u$  and node  $v$ . Node  $u$  is allowed to transmit a combination of packets that belongs to the generation if  $u$  carries a token number greater than 2. After the transmission, node  $u$  decrements its token number by 1 and the two nodes redistribute their token numbers, so that the sum of the two nodes' token numbers is reallocated to the two nodes in proportion to their ranks, i.e., the number of linearly independent packets carried by the nodes<sup>9</sup> A node carrying a token value less than 2 can only transmit combinations of packets that belong to the generation to the destination. If the initial number of tokens for the generation is  $L_g$ , one can show that the total number of combinations transmitted for the generation is bounded by  $L_g$ . The actual number of combinations being transmitted is usually smaller than  $L_g$  when a recovery scheme is employed.

A different replication control approach for RLC, called *E-NCP* (Efficient Protocol based on Network Coding), was proposed in [31]. Under the E-NCP protocol, to transmit  $K$  source packets from a source node to a destination node, the source node generates  $K'$  (slightly larger than  $K$ ) random linear combinations (which is referred to as pseudo source packets) from the  $K$  source packets, and disseminates these  $K'$  pseudo source packets to the first  $K'$  relay nodes it meets respectively<sup>10</sup>. Each of the  $K'$  relay nodes subsequently uses binary spray-and-wait mechanism to limit the total number of transmissions

---

<sup>9</sup> Two nodes with no information to exchange also re-allocate their token numbers in proportion to their ranks, the rationale being to keep the potential of a node to spread information of the generation proportional to its rank.

<sup>10</sup>The reason for disseminating  $K'$  pseudo source packets is so that the original  $K$  source packets can be decoded with high probability when  $K$  coded-packets are received. As [57] pointed out, it suffices for the source to disseminate to the  $K$  relay nodes the original  $K$  source packets or  $K$  linearly independent coded-packets.

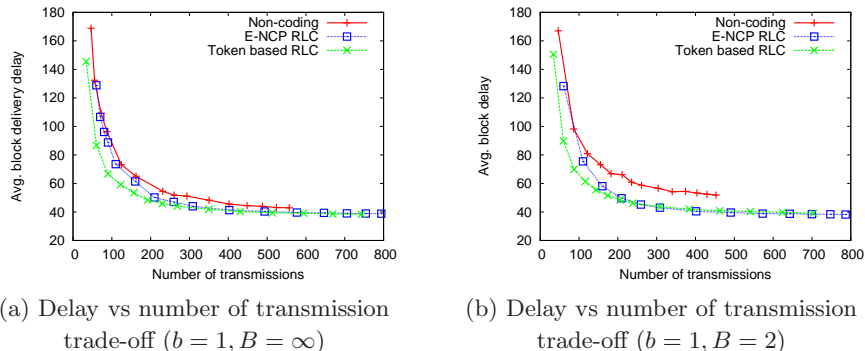


Figure 7: Block delivery delay vs transmission number trade-off under same network setting as Fig. 5 except for the bandwidth and buffer constraints

made for its pseudo source packet. Different pseudo source packets can then be combined at intermediate nodes (see details in [31]).

Simulation studies reported in [57] compared the block 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. 7 plots the average block delivery delay versus number of transmissions, for a block of  $K = 10$  packets, under different token limits, for the cases both without buffer constraints (a) and with buffer constraint of  $B = 2$  (b). We observe that, with a similar number of transmissions, the RLC schemes achieve smaller block delivery delay than non-coding schemes, and the token-based RLC scheme outperforms the E-NCP scheme, especially for small numbers of transmissions. The results for limited relay buffer case further establish the benefits of the RLC schemes in reducing block delivery delay without increasing transmission overhead.

The fact that the RLC scheme improves the delivery delay versus the number of transmissions trade-off explains the benefits observed for RLC scheme under multiple generation case [55, 57]. Consider a network scenario where there are multiple continuous unicast flows, each source independently generates blocks of packets according to a Poisson process, and RLC is applied to packets belonging to the same block, i.e. each block forms a generation. Whether network coding is used or not, replication control mechanism, by limiting the number of transmissions made, can reduce bandwidth contention and in turn reduce the system-wide average block delivery delay. There exists an optimal token value under which the average block delivery delay is minimized: a too large token value or no replication control lead to severe contention in the network and degraded performance, while a too small token value prevents network nodes from exploiting all available bandwidth. In [55, 57] it has been shown that the minimum delivery delay achieved by the RLC scheme is smaller than that achieved by non-coding scheme under various traffic rates, especially when the buffer space is constrained.

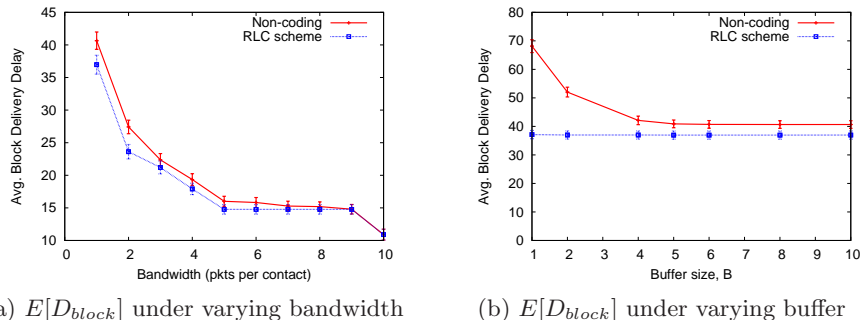


Figure 8: Impact of bandwidth and buffer constraints under same network setting as Fig. 5 except the bandwidth and buffer constraints for (a) and (b) respectively

### 5.3. Discussion about RLC benefits

In this section, we summarize the simulation studies reported in [55, 57] that illustrate the impact of different system parameters on the benefit of RLC scheme.

#### 5.3.1. Impact of Different Bandwidth and Buffer Constraints

For DTNs with bandwidth and buffer constraints, we have seen that the RLC scheme achieves much smaller block delivery delay than the non-coding scheme. As the network bandwidth becomes less and less constrained, the RLC benefit diminishes and becomes non-existent when the number of packets that can be exchanged during each contact,  $b$ , equals to the block size  $K$ . In this case, the  $K$  packets propagate independently without competing for bandwidth<sup>11</sup>. For example, Fig. 8(a) from [57] plots the average block delivery delay and its 95% confidence interval (based on 50 different simulation runs) under varying bandwidth constraints, for the case where there are  $k = 10$  packets from the same unicast flow propagating in a network of 101 nodes. Likewise, the improvement on the delay versus transmissions trade-off achieved by the RLC scheme over the non-coding scheme diminishes when bandwidth becomes less constrained.

The RLC scheme is especially beneficial for the case where, in addition to the bandwidth constraint, the buffer space at relay nodes is also constrained. Fig. 8(b) plots the average block delivery delay (and the 95% confidence interval) for a block of  $K = 10$  packets achieved by the RLC scheme and the non-coding scheme under varying nodal buffer sizes,  $B (B \leq K)$ . We observe that, as the buffer space becomes more and more constrained, the performance under the RLC scheme only degrades slightly, in sharp contrast to the non-coding scheme. As different packets are mixed randomly under the RLC scheme, dropping a combination has same effects to all packets encoded in the combination. Therefore, the RLC scheme allows an even propagation of different information in

<sup>11</sup>Therefore, the block delivery delay coincides with the epidemic routing delay under no resource constraints as characterized in [56].

the network. For the non-coding scheme, when a block of packets starts being spread in the network, small differences in the number of copies are amplified: the more copies a packet has in the network, the more this packet is copied to other nodes<sup>12</sup>. 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 block of packets.

### 5.3.2. Impact of Generation Management

We now discuss the impact of generation management, i.e., the decision on how many and which packets form a generation to which random linear coding operations are applied to.

So far, we have focused on the case where a generation is formed by packets from the same unicast flow. Other ways to form a generation have been explored in [55, 57], including the case where multiple packets from different sources but destined to the same destination form a generation, and the case where multiple packets from different sources destined to different destinations form a generation. We refer to these three cases respectively as *SS\_SD*, *MS\_SD* and *MS\_MD* (Single/Multiple Source, Single/Multiple Destination). The benefit achieved by the RLC scheme for the *MS\_SD* case is smaller than for the *SS\_SD* case. Basically, under the *MS\_SD* case, the  $K$  packets start to propagate from the  $K$  different source nodes, and under non-coding schemes the effect of relay nodes choosing the wrong packets to forward becomes less significant. For the *MS\_MD* case, the RLC scheme performs worse than the non-coding scheme as mixing packets destined to different destinations forces every destination node to receive  $K$  independent combinations to decode the one single packet destined to it. In general, RLC scheme should restrict mixing to packets that are destined to the same destination.

References [55, 57] also show that as the generation size,  $K$ , increases, the relative benefit of the RLC schemes decreases. This is because for the non-coding scheme, with a larger block size, there are a larger number of packets to randomly choose from, and therefore the probability of two paths choosing to forward the same packet is smaller.

### 5.3.3. Impact of Control Signaling

All above results for unicast application are for the *full signaling* case, where two encountering nodes first exchange information about which packets (or coded packets) they each carry in order to avoid transmitting useless information to each other. However, the exchange of this control information consumes transmission bandwidth, especially for RLC schemes, where more information is exchanged (the encoding coefficients for each coded-packet compared to the

---

<sup>12</sup> As a first approximation the spread rate of a packet with  $n$  copies in a network with  $N$  nodes is proportional to  $n(N - n)$ , then the packet with a larger number of copies spread faster as long as  $n < N/2$ .



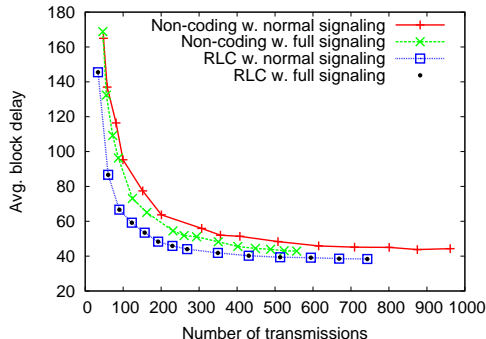


Figure 9: Block delivery delay vs transmission number trade-off with full signaling and normal signaling, the network setting is the same as that of Fig. 5

packet ID for each packet). Also, for RLC schemes full signaling incurs computation overhead, because each node needs to perform some calculations in order to determine whether it has useful information for the other on the basis of the received encoding coefficients.

Now let us consider *normal signaling*, where nodes do not exchange information about the packets or coded-packets they carry. Without such information, under the non-coding scheme, a node randomly chooses a packet from the set of packets it carries, and forward it to the other node; under the RLC scheme, a node generates and transmits a random linear combination to the other node as long as the other node has not reached full rank yet. Fig. 9 plots the block delivery delay versus the number of transmissions trade-off achieved by the non-coding and the RLC scheme under full signaling and normal signaling. We observe that the performance of the RLC scheme under normal signaling is almost identical with that under full signaling, whereas for the non-coding scheme, the performance under normal signaling is significantly worse than that under full signaling. This demonstrates that the RLC scheme retains its benefit over the non-coding scheme even with less signaling overhead, i.e., when normal signaling is used for RLC scheme while full signaling is used for the non-coding scheme.

#### 5.4. Modeling Studies of Network Coding Scheme

In order to quantitatively analyze the benefit of RLC schemes, an Ordinary Differential Equations (ODEs) based model has been proposed to characterize the block delivery delay under the non-coding and RLC scheme in [33, 32].

The model characterizes the propagation of a block of  $K$  packets that are generated by the source at time  $t = 0$ , all destined to the same destination node in a network with  $N + 1$  mobile nodes. The inter-contact time between each pair of nodes follows an exponential distribution with rate  $\lambda$ . During each contact, one packet can be transmitted between two nodes in each direction. Each node has limited buffer space and can store at most  $B$  ( $B < K$ ) relay packets. For the

related problem of gossip algorithm for spreading multiple rumors or messages to the whole network, many efforts have been focused on characterizing the stopping time, i.e., the number of time slots by which all nodes receive all messages achieved by different gossip algorithm ([37] and references therein).

Under simplifying assumptions to be explained later, for both the RLC scheme and the non-coding scheme, the system state at time  $t$  can be described by a  $B$ -tuple  $\{X_1(t), X_2(t), \dots, X_B(t)\}$ , where  $X_i(t)$  denotes the number of rank  $i$  nodes in the network (excluding the source node). The number of nodes with rank zero is given by  $X_0(t) = N - \sum_{j=1}^B X_j(t)$ .

Consider the RLC scheme with normal signaling<sup>13</sup>, and further assume that, when a node receives a random linear combination from another node, its rank always increases by 1, i.e., the combination contains useful information for the node. This is an optimistic assumption, since the other node might have no useful information for the node<sup>14</sup>, and even the other node has useful information for the node, the random linear combination it generated might contain no useful information for the node. The following ODEs model results:

$$\frac{dX_1(t)}{dt} = \lambda \left( \sum_{j=1}^B X_j(t) + 1 \right) X_0(t) - \lambda \sum_{j=1}^B X_j(t) X_1(t) \quad (1)$$

$$\frac{dX_i(t)}{dt} = \lambda \sum_{j=1}^B X_j(t) X_{i-1}(t) - \lambda \sum_{j=1}^B X_j(t) X_i(t) \quad (2)$$

for  $i = 1, \dots, B - 1$ ,

$$\frac{dX_B(t)}{dt} = \lambda \sum_{j=1}^B X_j(t) X_{B-1}(t) \quad (3)$$

with initial conditions  $X_i(t) = 0$  for  $i = 1, \dots, B$ . These ODEs characterize the changing rates of  $X_i(t)$ ,  $i = 1, 2, \dots, B$ . For example,  $X_1(t)$  increases by one whenever a rank 0 node encounters a node with rank greater than 0 (there are a total of  $\sum_{j=1}^B X_j(t) + 1$  such nodes including the source node), therefore  $X_1(t)$  increases with rate  $\lambda(\sum_{j=1}^B X_j(t) + 1)X_0(t)$ . On the other hand,  $X_1(t)$  decreases by one whenever a rank 1 node increases its rank by 1 by meeting some node (including the source node, but rather than itself) with rank greater than 0 (there are a total of  $\sum_{j=1}^B X_j(t)$  of such nodes), therefore the decrement rate of  $X_1(t)$  is  $\lambda \sum_{j=1}^B X_j(t) X_1(t)$ . Considering both increasing and decreasing rates of  $X_1(t)$  yields (1).

Let  $T_i$ ,  $i = 1, \dots, K$  denote the time instant when the destination node reaches rank  $i$ . The block delivery delay is then  $T_K$ . The following ODEs hold for the

---

<sup>13</sup>This means that when two nodes,  $u$  and  $v$ , encounter each other, node  $u$  transmits a random linear combination to node  $v$  as long as  $u$  has rank greater than 0 and  $v$  has not reached full rank yet (the transmission from node  $v$  to node  $u$  is similar).

<sup>14</sup>i.e., all the coded packets at another node are linear combinations of the coded packets at the node

cumulative distribution function of  $T_i$ ,  $F_i(t) := \Pr(T_i < t)$ :

$$\begin{aligned}\frac{dF_1(t)}{dt} &= \lambda \left( \sum_{j=1}^B X_j(t) + 1 \right) (1 - F_1(t)) \\ \frac{dF_i(t)}{dt} &= \lambda \sum_{j=1}^B X_j(t) (F_{i-1}(t) - F_i(t)), \text{ for } i = 2, \dots, K,\end{aligned}$$

with initial conditions given by  $F_i(0) = 0$ , for  $i = 1, 2, \dots, K$ .

For the non-coding scheme, in order to characterize the probability that two encountering nodes have useful information (i.e., new packets) for each other, it is assumed that the  $i$  packets (with  $i = 1, \dots, B$ ) carried by a rank  $i$  node have been drawn with equal probability from the original set of  $K$  packets. As a result,  $\Pr(i, j)$ , the probability that a node carrying  $i$  packets can receive a new packet from a node carrying  $j$  packets, can be expressed as

$$\Pr(i, j) = \begin{cases} 1, & \text{if } i < j, \\ 1 - \binom{i}{j} / \binom{K}{j}, & \text{if } i \geq j. \end{cases}$$

The ODEs in Eq(1-3) can then be modified by introducing  $\Pr(i, j)$  in order to model the non-coding scheme. Basically, the transition rates are ‘‘thinned’’ as the following example demonstrates:

$$\frac{dX_i}{dt} = \lambda \sum_{j=1}^B \Pr(i-1, j) X_j(t) X_{i-1}(t) - \lambda \sum_{j=1}^B \Pr(i, j) X_j(t) X_i(t) \text{ for } i = 1, \dots, B-1$$

The above ODEs are solved numerically to obtain the average block delivery delay of both non-coding scheme and coding scheme and compared against simulation studies in [33, 32]. Three different scheduling policies for the non-coding scheme are simulated: *random policy*, *local rarest policy*, and *global rarest policy*. The average block delivery delay under the simulation is then compared against those predicted by the ODE models, as shown in Fig. 10, which plots the block delivery delay under varying  $N$ , the total number of nodes in the network excluding the source node. We observe that the ODE models under-estimate the block delivery delay for both the RLC scheme and non-coding scheme, an expected result considering the optimistic assumptions made. Among the three scheduling policies, global rarest policy, an oracle scheme assuming global knowledge about the number of copies each packet has in the network, performs the best, followed by the random policy, and the local rarest policy performs worst. The simulation results also show that the RLC scheme, which does not rely on any oracle, performs as well as the non-coding scheme with global rarest policy, an oracle-based scheme.

##### 5.5. Other Works on RLC scheme in Unicast Application

In this section, we briefly review other research efforts that apply network coding to unicast application in DTNs.

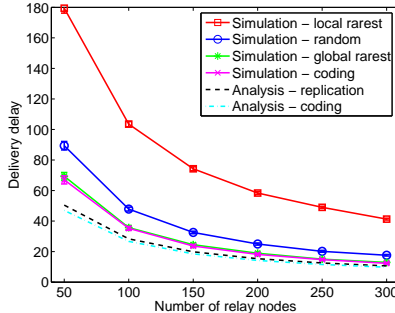


Figure 10: Block delivery delay under different number of nodes (Fig. 3 in [31])

### 5.5.1. Priority Coding Protocol

We have seen that the RLC scheme performs better than non-coding scheme in terms of time to deliver the last packet in the block, i.e., block delivery delay, but performs worse in terms of average packet delivery delay (Fig. 5(b)). For applications where different packets have different priorities, [31] proposes a priority coding protocol which works as follows. Suppose that the source generates  $K$  packets destined to the same destination node, and the  $K$  packets can be classified into  $M$  ( $M < K$ ) priority levels, so that the packets of level  $i$  should be decoded before packets of level  $j$ , if  $i < j$ . The source node first transmits packets from level 1 using the RLC scheme, applying coding to all packets of this level. Only after receiving the ACK for the level 1 packets (generated by the destination and propagated by all nodes under VACCINE recovery scheme), does the source start to transmit packets from level 2 using the RLC scheme. Nodes in the network drop locally stored combinations of lower level packets when they receive combinations of higher level packets. The source starts to transmit packet of level  $i + 1$  ( $i = 1, 2, \dots, M - 1$ ) upon receiving ACK for packets of level  $i$ , until all  $K$  packets are decoded by the destination.

Using the ODE models in Section 5.4, [32] found that the overhead of this priority coding protocol, in terms of the increase in the time to deliver all the  $K$  packets as compared to non-priority RLC scheme, increases linearly with the number of priority levels. For each priority level, the priority coding protocol incurs an extra delay equal to the time to deliver the first packet of the level and the time to propagate ACK from the destination to the source.

### 5.5.2. Optimal Control of Two-Hop Scheme

In [2], the authors consider the case where  $K$  packets are generated at a source (at different time instances) and need to be delivered to a destination by a deadline. The source can perform random linear combinations of the packets before forwarding information to relay nodes. Each relay node can store at most one linear combination, and only transfers the (coded) packet it carries to the destination. The purpose is to maximize the delivery probability of the  $K$  packets by the given deadline. The authors show that the best strategy for

the source when meeting a relay is to combine all the packets available at that time. A lower bound for the delivery probability is also derived.

We comment that this routing scheme is essentially an RLC scheme with 2-hop routing where each packet or coded-packet can traverse a path of one or two hops to reach the destination. As only the source node performs random linear coding, it is not “network coding” in the strict sense.

### 5.5.3. Network Coding Based Secure Communication for DTN

For DTNs with unidentified Byzantine adversarial nodes that introduce corrupted packets into the network, a RLC based secure unicast routing protocol has been proposed in [42].

Consider a block of  $K$  packets generated at a source node and destined to a destination node. Suppose that the adversarial nodes can pollute the entire coded packets, i.e., including the encoding coefficients in the headers. The destination node might not be able to decode the  $K$  packets with any  $K$  independent coded-packets, as some of them might be polluted.

The proposed protocol combines cryptographic key dissemination and error-correction coding together to ensure secure data transmission. The source node encodes the  $K$  data packets into the error-correction coded vector space [27], and then generates private and public keys from the coded-packets [58]. The public keys are then distributed to a certain fraction of nodes in the network. Nodes that have received the public keys are secured, as they can use the public keys to verify the received linear combinations. Finally, the source node sends the coded-packets using the RLC scheme, adopting the E-NCP replication control mechanism.

For the simple setting of a single generation and no bandwidth or buffer constraints, the analysis of the above protocol [42] characterizes the probability that the destination node successfully decodes the  $K$  packets within a certain time, in terms of protocol parameters such as the dimension of the error-correction coded vector space, the number of nodes that needs to receive the public keys and the binary spray-and-wait counter.

## 6. Open Issues

In this section, we discuss several open issues that deserve further investigations.

### 6.1. RLC Benefits for Application with Short Messages

Until now, most existing works have applied random linear coding to a block of  $K$  packets that are generated at the source node in a batch, and have considered the time to deliver the whole block of packets (i.e., the block delivery delay). The batch arrival of multiple packets can be due to the applications generating long messages, e.g. in the case of file sharing. The long messages need to be fragmented into smaller packets, in part to take advantage of the often short transmission opportunities.

For applications that generate short messages, it is inefficient to fragment the short message into very small packets. More packets to send means more lower layer protocol overhead. Furthermore, the relative overhead of the RLC scheme (to store and transmit coding coefficients) as given by  $K \log_2(q)/P$  is larger for shorter messages, i.e., for small values of  $P$ .

Whether and when random linear coding, or more generally network coding, is beneficial in this case remains an open question. Instead of block delivery delay, average packet delivery delay or in-order delivery delay should be considered, given that the packets are independent from each other. The central question to the operation of RLC scheme is the generation forming, i.e., how to decide which packets form a generation to which RLC is applied. Above, we have seen that combining packets belonging to different unicast flows results in smaller benefits (nevertheless, still beneficial in the buffer constrained case). It remains to be studied the options of combining packets (from the same unicast flow) generated at different times.

### 6.2. RLC scheme for Multicast Communication

Multicast communication supports the distribution of a data packet to a group of users. The application of network coding to multicast communication in DTNs has received less attention than that of broadcast and unicast, partly due to the different semantics and multicast algorithms for multicast in DTNs [59, 54, 1, 6].

For DTN based content-distribution network with bandwidth constraint, reference [23] proposed to adapt network coding to the different social interests of users and only mix contents of similar type. Preliminary simulation results show that network coding based scheme achieves lower decoding delay than non-coding scheme. Moreover, when network coding is only applied to messages of the same type, users that are only interested in a small set of contents experiences smaller delay, at the cost of larger delay of users with wide interests. To trade-off between the potential throughput improvement and increased delay of network coding based scheme, the authors proposed to adjust the mixing so that contents of different types but with a large overlap in their receivers can be mixed.

Given that RLC schemes have shown better performance in the two extreme cases of multicast communication, i.e., broadcast and unicast, some results reviewed in this chapter might also apply to multicast communication. Further works need to be done to fully understand the network coding's benefit for multicast communication in DTNs.

## 7. Summary and Conclusions

In this chapter, we review existing research that applies network coding to Delay Tolerant Networks (DTNs).

We first provide a background on traditional non-coding based routing schemes for broadcast and unicast applications in DTNs and on the basic operation of

Random Linear Coding (RLC), the form of network coding usually considered in literature. Then we introduce the design space for DTN routing, emphasizing the different options to integrate network coding. The main focus of the chapter is on the performance evaluation of RLC based routing schemes both for broadcast and unicast communications, the second ones having been the object of a larger amount of research. We highlight both theoretic results and simulation studies findings. For both communication models, the RLC based scheme provides better trade-off between energy consumption and delivery performance.

Beyond the above results, we also review various research efforts that extend the basic RLC schemes for example to achieve priority routing and secure communication. Finally, we discuss open issues that require in our opinion further investigation.

## References

- [1] M. Abdulla and R. Simon. A Simulation Analysis of Multicasting in Delay Tolerant Networks. In *Conference on Winter simulation (WSC)*, Monterey, CA, USA, 2006.
- [2] E. Altman, F. De Pellegrini, and L. Sassatelli. Dynamic Control of Coding in Delay Tolerant Networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, Shanghai, China, 2010.
- [3] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, 2007.
- [4] 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.
- [5] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. In *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Network Research: Research, Trends and Applications*, 2002.
- [6] M. C. Chuah and Y. Xi. An Encounter-based Multicast Scheme for Disruption Tolerant Networks. Technical Report LU-CSE-07-009, Lehigh University, 2007.
- [7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, 1996.
- [8] First Mile Solutions (i.e., DakNet). <http://firstmilesolutions.com/>.
- [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 (dyd)*, Bangalore, India, Dec 2002.



- [11] C. Fragouli, J. Widmer, and J.-Y. Le Boudec. Efficient Broadcasting Using Network Coding. *IEEE/ACM Transactions on Networking*, 16(2):450–463, April 2008.
- [12] R. Groenevelt, P. Nain, and G. Koole. The Message Delay in Mobile Ad Hoc Networks. *Performance Evaluation*, 62(1-4):210–228, October 2005.
- [13] Z. Haas, J. Halpern, and L. Li. Gossip-based Ad Hoc Routing. In *IEEE International Conference on Computer Communications (INFOCOM)*, New York, NY, USA, 2002.
- [14] Z. J. Haas and T. Small. A New Networking Model for Biological Applications of Ad Hoc Sensor Networks. *IEEE/ACM Transactions on Networking*, 14(1):27–40, February 2006.
- [15] B. Haeupler. Analyzing Network Coding Gossip Made Easy (simpler proofs for stronger results even in adversarial dynamic networks). In *Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, 2010.
- [16] D. Hay and P. Giaccone. Optimal routing and scheduling for deterministic delay tolerant networks. In *International Conference on Wireless On-Demand Network Systems and Services (WONS)*, Snowbird, Utah, USA, 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)*, Yokohama, Japan, June-July 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, 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)*, New York, NY, USA, 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*, pages 1973–1982, 2005.
- [22] 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)*, San Jose, CA, USA, 2002.
- [23] G. Karbaschi and A. C. Viana. A Content-based Network Coding to Match Social Interest Similarities in Delay Tolerant Networks. In *First Extreme Workshop on Communication (ExtremeCom)*, Laponia, Sweden, 2009.

- [24] S. Katti, D. Katabi, W. Hu, and R. Hariharan. The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments. In *Allerton Conference on Communication, Control, and Computing*, Montecello, IL, USA, 2005.
- [25] 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(3):497–510, 2008.
- [26] 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 2000*, volume 64, 2002.
- [27] R. Koetter and F. Kschischang. Coding for Errors and Erasures in Random Network Coding. *IEEE Transactions on Information Theory*, 54(8):3579–3591, August 2008.
- [28] A. Krifa, C. Barakat, and T. Spyropoulos. Optimal buffer management policies for delay tolerant networks. In *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, San Diego, CA, USA, 2007.
- [29] R. Lidl and H. Niederreiter. *Finite Fields, 2nd edition*. Cambridge, England: Cambridge University Press, 1997.
- [30] Y. Lin, B. Li, and B. Liang. Differentiated Data Persistence with Priority Random Linear Codes. In *International Conference on Distributed Computing Systems (ICDCS)*, Toronton, Canada, 2007.
- [31] Y. Lin, B. Li, and B. Liang. Efficient network coded data transmissions in disruption tolerant networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, 2008.
- [32] 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, June 2008.
- [33] Y. Lin, B. Liang, and B. Li. Performance modeling of network coding in epidemic routing. In *ACM/SIGMOBILE International Workshop on Mobile Opportunistic Networking (MobiOpp)*, Puerto Rico, 2007.
- [34] J. Liu, D. Goeckel, and D. Towsley. Bounds on the gain of network coding and broadcasting in wireless networks. In *IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, 2007.
- [35] 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)*, Parma, Italy, 2004.
- [36] D.S. Lun, N. Ratnakar, M. Mdard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-Cost Multicast over Coded Packet Networks, journal=.
- [37] C. Avin M. Borokhovich and Z. Lotker. Tight Bounds for Algebraic Gossip on Graphs. *Arxiv preprint arXiv:1001.3265 v1 [cs. IT]*, 2010.

- [38] A. Maffei, K. Fall, and D. Chayes. In *Ocean Instrument Internet: Using Disruption Tolerant Networking to Join Heterogeneous Oceanographic Instrumentation into a Single Network*, Honolulu, HI, USA, 2006.
- [39] M. Mitzenmacher and E. Upfal. *Probability and computing: randomized algorithms and probabilistic analysis*. Cambridge Press, 2005.
- [40] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *ACM International Conference on Mobile Computing and Networking (Mobicom)*, Seattle, WA, USA, 1999.
- [41] J. Partan, J. Kurose, and B. N. Levine. A Survey of Practical Issues in Underwater Networks. In *ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, USA, 2006.
- [42] L. Sassatelli and M. Medard. Network coding for delay tolerant networks with byzantine adversaries. 2009.
- [43] T. Small and Z. J. Haas. The Shared Wireless Infostation Model - a new ad hoc networking paradigm. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2003.
- [44] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
- [45] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case. In *ACM/IEEE Transactions on Networking*, volume 16, pages 77–90, 2008.
- [46] 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, 2008.
- [47] 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.
- [48] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, April 2000.
- [49] J. Widmer and J.-Y. Le Boudec. Network Coding for Efficient Communication in Extreme Networks. *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
- [50] 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)*, Riva del Garda, Italy, 2005.
- [51] Wizzy Digital Courier. <http://www.wizzy.org.za/>.
- [52] 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.

- [53] 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.
- [54] Q. Ye, L. Cheng, M. C. Chuah, and B. D. Davison. OS-multicast: On-demand Situation-aware Multicasting in Disruption Tolerant Networks. In *IEEE Vehicular Technology Conference (VTC)*, Montreal, Canada, 2006.
- [55] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the Benefits of Random Linear Coding for Unicast Applications in Disruption Tolerant Networks. In *NETCOD (IEEE Workshop on Network Coding, Theory, and Applications)*, 2006.
- [56] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance Modeling of Epidemic Routing. *Elsevier Computer Networks journal*, 51/10:2859–2891, 2007.
- [57] 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/>.
- [58] F. Zhao, T. Kalker, M. Medard, and K. J. Han. Signatures for content distribution with network coding. In *IEEE International Symposium on Information Theory (ISIT)*, Nice, France, 2007.
- [59] W. Zhao, M. Ammar, and E. Zegura. Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms. In *ACM Sigcomm workshop on Delay-Tolerant Networking (WDTN)*, Philadelphia, PA, 2005.