Data Exchange in Delay Tolerant Networks using Joint Inter- and Intra-Flow Network Coding

Pouya Ostovari
dept. of Technology
San Jose State University
San Jose, USA
pouya.ostovari@sjsu.edu

Jie Wu
dept. of Computer & Inf. Sciences
Temple University
Philadelphia, USA
jiewu@temple.edu

Alireza Jolfaei
School of Science, Eng. & Inf. Technology
Federation University Australia
Mount Helen, Australia
a.jolfaei@federation.edu.au

***Abstract*—Data transmission in delay tolerant networks (DTNs) is a challenging problem due to the lack of continuous network connectivity and nondeterministic mobility of the nodes. Epidemic routing and spray-and-wait methods are two popular mechanisms that are proposed for DTNs. In order to reduce the transmission delay in DTNs, some previous works combine intra-flow network coding with the routing protocols. In this paper, we propose two routing mechanisms using systematic joint inter- and intra-flow network coding for the purpose of data exchange between the nodes. We discuss the reasons why inter-flow network coding helps to reduce the delivery delay of the packets, and we also analyze the delays related with only using intra-flow coding, and joint inter- and intra-flow coding methods. We empirically show the benefit of joint coding over just intra-flow coding. Based on our simulation, joint coding can reduce the delay up to 40%, compared to only intra-flow coding.**

***Index Terms*—Data exchange, delay tolerant networks, inter-flow coding, intra-flow coding, random linear network coding.**

I. Introduction

A delay tolerant network (DTN) [1] is a type of wireless mobile network where a contemporary path may not exist between a pair of source and destination nodes. Network partitioning is frequent in these networks, which can be due to high mobility, low density, short radio range, or obstacles. Because of frequent partitioning and no infrastructure support, conventional routing protocols do not perform well in DTNs. Epidemic routing has been proposed for routing in DTNs, which adopts a store-carry-forward paradigm. In epidemic routing, the received packets are buffered and carried by the nodes. The nodes pass the buffered packets to the new nodes when they encounter one another.

Recently, the benefits of linear network coding (NC) in routing protocols for DTNs have been investigated [2]–[5]. In NC [6]–[10], the source transmits coded packets in the form of $\sum_{i=1}^{m} \alpha_i \times P_i$, where P and α, are the original packets and random coefficients, respectively. The relay nodes perform the same operation on the received packets. The destination can use Gaussian elimination to decode the packets once it receives m linearly independent coded packets. NC can increase the throughput of wired and wireless networks [6], [8]. Because of low density of the nodes, the benefit of NC in increasing the network throughput of DTNs is negligible. However, the limited buffer and bandwidth in DTNs creates new opportunity for NC in decreasing the delivery delay of the packets.

This research was supported in part by NSF grants CNS 1757533, CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, and IIP 1439672.

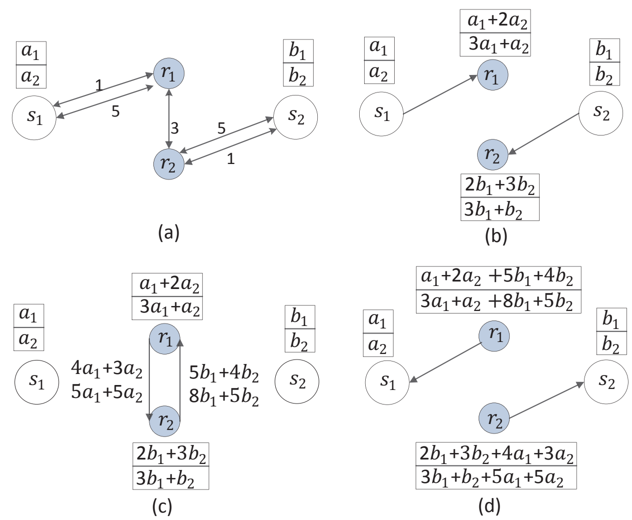


Fig. 1. Inter- and intra-flow NC. (a) Temporal network representation, (b) Time-slot 1 , (c) Time-slot 3, (d) Time-slot 5.

Network coding can be classified into *intra-flow* and *inter-flow*. In intra-flow NC, the packets from the same flow are coded together. In contrast, inter-flow NC codes the packets of different flows together. The authors in [3], [11] investigated the benefit of intra-flow NC for unicast applications, and showed that the intra-flow NC-based forwarding method achieves the minimum delivery time. When NC is not used in DTNs, the destination node might receive some of the packets multiple times, and might not receive the rest of the packets, which is known as coupon collector problem. However, when network coding is applied, every coded packet has the same importance, and contributes the same amount of data to the destination. Using NC, with a high probability every received coded packet by the destination node is linearly independent to the packets in the destination node's buffer [6]; as a result, NC solves the coupon collector's problem in DTNs, and reduces the delivery delay.

In [3], the authors also studied inter-flow NC in the case of multiple flows with different destination nodes, and showed that inter-flow NC increases the delivery time. The reason is

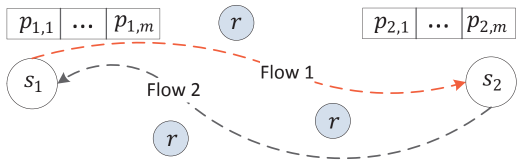


Fig. 2. Network model.

that when packets destined to two different destination nodes are coded together, each destination node needs to receive more coded packets to be able to decode its own packets, as well as the other destination nodes' packets. Our goal in this paper is to show that depending on the application, inter-flow NC can be useful as well. That happens when the source of a set of packets is the destination of the other source's packets. In a military ad hoc network application, assume there are several isolated storage centers (also called dropboxes, or sources, here). These static storage centers need to exchange data for replication and/or consistency purposes. This can be done through mobile relays (tanks or soldiers) when they visit these centers. The first reason that joint inter- and intra-flow NC reduces the delay in this kind of application is that it solves the bottleneck problem. Moreover, as the movements and contacts in DTNs are unpredictable, finding the optimal routing that results in the minimum delivery delay is not possible without NC. Joint inter- and intra-flow NC helps us to achieve the optimal delivery delay using a simple scheme.

Assume that the buffer size of the relay and the source nodes in Fig. 1 are 2 and 4 packets, respectively. Moreover, the bandwidth of each node is 2 packets, which means each node can transmit two packets to its neighbor. The temporal network representation of a DTN is shown in Fig. 1(a). At time-slot 1, nodes r_1 and r_2 encounter the sources s_1 and s_2, respectively. The sources transmit 2 intra-flow coded packets to the relay nodes. The relay nodes meet each other at time-slot 3. Without having a global knowledge about the future contacts, the nodes cannot discover that exchanging their packets results in the optimal delay. Assume that each relay node passes one coded packet and the nodes assign one location of their buffer to each flow. Thus, with only intra-flow NC, the destinations will receive one coded packet from the opposing flow in time-slot 5, so they are no able to decode the coded packet. However, with inter-flow NC, the nodes simply exchange their packets, and code the received packets with the packets in their buffers. The relay nodes meet the sources for the second time at time-slot 5, and transmit the packets to the destinations (sources). In this case, the destinations have 4 linearly independent packets, and are able to decode the original packets.

In this paper, we study the advantage of systematic joint inter- and intra-flow NC for data exchange in DTNs. We also analyze the delivery delay of the packets in the case of using only intra-flow NC, and joint inter- and intra-flow NC. Moreover, we confirm our analysis by comparing our findings with simulation results. We evaluate our method through simulations on both synthetic data and real data trace. Our result can be applied in data-centric ad hoc mobile environments for efficient data dissemination among data storages.

The remainder of this paper is organized as follows: Section II provides the necessary background about NC and DTNs. In Section I, we define the problem and the settings. In Section IV, we propose our joint inter- and intra-flow NC methods, and analyze the delivery delay in different scenarios. We evaluate the proposed methods through simulations in Section V, and Section VI concludes the paper.

II. Related Work and Background

In random linear NC [6], which we call NC for simplicity, coded packets are random linear combinations of the original packets over a finite field. Using random linear NC, the source node generates and transmits random coded packets. The destination nodes are able to decode the coded packets once they receive m linearly independent coded packets. The decoding process is done using Gaussian elimination for solving a system of linear equations. In [6], it is shown that when the coefficients of the coded packets are selected randomly, with a high probability the received coded packets by the destination node are linearly independent.

Epidemic routing [12] is the simplest routing method in DTNs, in which the source node gives a copy of the packets to each encountered node. The relay nodes repeat the same process until the destination receives the packets, which incurs too many transmissions and copies of the same packets. In order to reduce the overhead of the epidemic routing, spray-and-wait method has been proposed in [13]. The spray-and-wait method limits the number of copies of each packet by assigning a number of tokens to each packet. The authors in [2] combine probabilistic forwarding with NC to perform a trade-off between the number of transmissions and the delay.

The work in [3] combines unicast epidemic routing and spray-and-wait methods with NC. In the proposed epidemic method, the source node transmits random linear coded packets to its neighbors, and the relay nodes repeat this process until the destination collects a sufficient number of coded packets. In the spray-and-wait method, the set of packets are assigned a given number of tokens. Once a node transmits coded packets to its neighbor, it shares its tokens with the neighbor, proportional to the rank the two nodes' buffer. The authors show that, in the case of multiple unicast flows with different destinations, inter-flow NC not only does not reduce the delay, but also increases it. The reason is that, when joint inter- and intra-flow NC is performed, each destination needs to receive all of the flows, which increases the number of required packets. In contrast, we can benefit from joint inter- and intra-flow NC in our methods, since, in our model, the source of a flow is the destination of the other flow.

III. Network Model

We consider a DTN with N source nodes that want to exchange their packets with the help of n relay nodes, as shown in Fig. 2. Each source node has a set of m packets to be transmitted to its destination, which is a source node, itself. We say that two nodes encounter (meet) when they are within the transmission range of each other. Two nodes can transmit

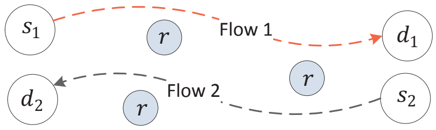


Fig. 3. Multiple flows with different destination nodes.

TABLE I
THE SET OF SYMBOLS USED IN THIS PAPER.

Sym.	Definition
s_i/r_i	The ith source/relay node
c_i	The coefficient matrix of the ith node
B/b	The buffer size of the relay node/ bandwidth limit
t_i	The tokens of the ith node
m	The number of source packets of each source node
g_i	Number of innovative packets of ith node to its neighbor
γ	Pairwise inter-meeting time of the nodes
λ	Average number of contacts of a node per time-slot
D_I^j	The delivery delay of intra-flow NC method with j flows
$D_{\hat{I}}$	The delivery delay of the joint inter- and intra-flow NC

data when they meet each other. We assume that the buffer size limit of the relay and the source (destination) nodes are equal to B and $2 \times m$ packets, respectively. Moreover, each node can transmit b packets to its neighboring node during their contact time. In other words, the graph of the network is directed, and the bandwidth (capacity) of each link is equal to b packets. Our objective is to exchange these packets between the sources with the minimum delivery delay, which is represented as D_I and, $D_{\hat{I}}$ in the case of only intra-flow NC, and joint inter- and intra-flow NC, respectively.

In the following sections we focus on the DTNs with $N = 2$, and our proposed algorithms and analysis are based on this assumption. However, the idea of our algorithms are general, and they can be easily applied to the DTNs with more than 2 source nodes. Table I summarizes the set of symbols used in this paper.

IV. ROUTING METHODS

It is shown in [3] that intra-flow NC can reduce the transmission delay of a single unicast flow in DTNs with limited bandwidth and buffers. The same work reveal the fact that inter-flow NC is not beneficial in the case of multiple unicast flows with different destinations (Fig. 3). Consequently, in our model, the destination of each flow is the source of the opposite flow; as a result, in the case of joint inter- and intra-flow NC, the destinations are not required to receive additional packets, as compared to the case of not using inter-flow NC. Thus, not only does inter-flow NC not have any negative impact on the delivery delay of the packets in our model, but it can also reduce it. In the following sections, we propose two routing methods with joint inter- and intra-flow NC.

A. Epidemic Routing with Joint Inter- and Intra-flow NC

We refer to our epidemic routing with joint Inter- and Intra-flow NC as EIINC. Assume that each source s_1 and s_2 has m packets to send. Each source performs intra-flow NC on its m packets and transmits packets in the form of $\sum_{i=1}^{m} \alpha_i \times P_i$ to its neighbors, where P and α are the packets and random coefficients, respectively. When two relay nodes encounter, they first exchange the coefficient vectors of the coded packets in their buffer to check whether they have innovative data for each other. An innovative packet to a node is a linearly independent packet to the packets in the node's buffer. If a node has innovative information for its neighbor, it transmits random linear combinations of the packets in its buffer, and the receiver mixes (codes) the received packets with the packets in its buffer. Once the received packet and a packet in the buffer of the receiver belong to two different flows, the coding results in an inter-flow NC. The number of packets transmitted by a node is equal to the minimum of the available bandwidth, and the number of innovative packets to its neighboring node. The processes at the source and relay nodes are shown in Algorithms 1 and 2, respectively.

Algorithm 1 The EIINC algorithm (source s_i side)

```
/* On encountering a node */
Run Algorithm 3
/*On receiving packets*/
Store the received packets in the buffer
```

Algorithm 2 The EIINC algorithm (relay r_i side)

```
/*On encountering a node*/
Run Algorithm 3
/*On receiving a packet*/
if buffer is full then
    Mix the packet with the buffered packets
else Store the packet in an empty location
```

Algorithm 3 On ith node encountering the jth node

```
Transmit the coefficients matrix, c_i
Receive the coefficients matrix of the neighbor, c_j
g_i = rank(c_i ∪ c_j) − rank(c_j)
if g_i > 0 then send min(g_i, b) random linear coded packets
```

The first benefit of joint inter- and intra-flow NC in DTN is simplifying the routing algorithm. In Fig. 1, 2 flows are initiated from the source nodes s_1 and s_2. In the first time-slot, the source nodes s_1 and s_2 transmit 2 intra-flow coded packets to nodes r_1 and r_2, respectively. The relay nodes r_1 and r_2 encounter at time-slot 3, and transmit 2 coded packets of the stored packets in their buffer. Each of the relay nodes code the received packets with the packets in their buffer, which is inter-flow coding. Later, the destination nodes meet the relay nodes at time-slot 5 and receive two joint inter- and intra-coded packets. Now consider the case of only intra-flow NC. When the relay nodes meet at time-slot 3, they should decide which packets should they store, as they cannot code packets from different flows together. Since the future contacts of the nodes are unpredictable, the nodes should randomly store some of the intra-coded packets. Assume that nodes r_1 and r_2 decide to keep one packet from each flow. In this case, the destination nodes receive one coded packet from the opposing flow, which

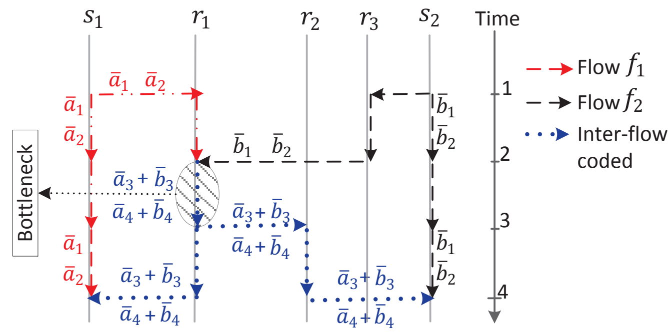


Fig. 4. Solving the bottleneck problem using joint inter- and intra-flow NC.

is not sufficient for encoding.

Joint inter- and intra-flow NC also solves the bottleneck problem. Consider source nodes s_1 and s_2 in Fig. 4, which want to exchange their packets $\{a_1, a_2\}$ and $\{b_1, b_2\}$, respectively. Assume that the buffer size of the relay nodes is equal to 2 packets. Let say $\bar{a}_i$ and $\bar{b}_i$ are random intra-coded packets over $\{a_1, a_2\}$ and $\{b_1, b_2\}$, respectively. The relays give 2 intra-coded packets to relays r_1 and r_3 at time-slot 1. At time-slot 2, the relay nodes r_1 and r_3 encounter one another. Without inter-flow NC, node r_1 cannot carry both flows. Thus, at least one of the destination nodes does not receive two coded packets from the opposing flow at time-slot 4, and is not able to decode the coded packets. As a result, the buffer of node r_1 becomes a bottleneck. With inter-flow coding, node r_1 can code the received packet from flow f_2 with the packet of flow f_1, which exists in its buffer. Using this approach, at time-slot 4 both the relay nodes r_1 and r_2 will have innovative packets for source nodes s_1 and s_2, respectively.

B. Delay Analysis

We first analyze the delivery delay of epidemic routing with intra-flow, and joint inter- and intra-flow NC, assuming the existence of k node disjoint paths. We prove that joint inter- and intra-flow NC reduces the delivery delay of two co-existing flows to the delivery delay in the case that just a single flow exists in the network. Then, we analyze the delay in the general case. For simplicity, in the rest of the paper, we refer to the epidemic routing with intra-flow, and epidemic routing with joint inter- and intra-flow NC, as intra-flow NC, and joint inter- and intra-flow NC, respectively.

1) k Disjoint Paths:

Theorem 1. *Assume that k node disjoint paths exist between two sources, that are destinations of each other. For $B \leq m$ and $b \leq m$, the delay of intra-flow NC is $2-\frac{n+k}{(n+k)\min(B,b)+m-1}$ times of the joint inter- and intra-flow NC.*

Proof. Consider the case of single flow with a buffer size and bandwidth equal to 1 packet. Assume that the number of relay nodes in the path is n nodes. The delivery delay of the first packet is $\gamma(n+1)$, where γ is the pairwise inter-meeting time of the nodes. The rest of the packets will be delivered in a pipeline fashion. Therefore, the total delay will be:

$$D_I^1 = \gamma(n+1) + \gamma(m-1) = \gamma(n+m)$$

In the case of k node disjoint paths with an average length of $\frac{n}{k}$ nodes, the delivery rate of the pathes are independent. As a result, the delay is:

$$D_I^1 = \gamma(\frac{n}{k}+1) + \gamma(\frac{m-1}{k}) = \gamma(\frac{n+m-1}{k}+1) \quad (1)$$

When the bandwidth and buffer size are smaller than m, the smaller one dominates the delivery rate of the packets. Consequently, for a buffer size and bandwidth equal to B and b packets, respectively, we can generalize Equation (1) to:

$$D_I^1 = \gamma(\frac{n}{k}+1) + \gamma(\frac{m-1}{k \times \min(B,b)}) = \gamma(\frac{n}{k} + \frac{m-1}{k \times \min(B,b)}+1)$$

When 2 flows share the same paths and relay nodes, the bandwidth and buffer of the relays are assigned between them evenly. Therefore, the buffer and bandwidth assigned to each of them is equal to $B/2$ and $b/2$, respectively. Consequently, we can modify Equation 1 to:

$$D_I^2 = \gamma(\frac{n}{k} + \frac{m-1}{k \times \min(\frac{B}{2}, \frac{b}{2})}+1) = \gamma(\frac{n}{k} + \frac{2(m-1)}{k \times \min(B,b)}+1) \quad (2)$$

When we use joint inter- and intra-flow NC, the whole bandwidth and buffer is shared between the 2 flows. It means that this case becomes similar to the case of intra-flows NC with a single flow. As a result, the delay of joint inter- and intra-flow NC is as follows:

$$D_{\hat{I}} = \gamma(\frac{n}{k} + \frac{m-1}{k \times \min(B,b)} + 1) \quad (3)$$

Therefore, from Equations (2) and (3) we have:

$$\begin{aligned}\frac{D_I^2}{D_{\hat{I}}} &= \frac{\gamma(\frac{n}{k} + \frac{2(m-1)}{k \times \min(B,b)} + 1)}{\gamma(\frac{n}{k} + \frac{m-1}{k \times \min(B,b)} + 1)} \\ &= 1 + \frac{(m-1)/(k \times \min(B,b))}{n/k + (m-1)/(k \times \min(B,b)+1)} \\ &= 2 - \frac{n+k}{(n+k)\min(B,b)+m-1} \quad (4)\end{aligned}$$

■

Based on Equation (4), as m goes to infinity, delivery delay of only intra-flow NC becomes 2 times that of the joint inter- and intra-flow NC methods. However, a large segment size makes NC impractical, due to the coding and decoding complexity. That is why for a large number of packets, the packets are partitioned into a set of segments, and coding is performed within each segment.

Theorem 2. *Consider a given configuration with a single flow f_1. Suppose that intra-flow NC results in delay D_I^1 for flow f_1. For a given and fixed configuration, and co-existing flows f_1 and f_2, joint inter and intra-flow NC results in delay $D_{\hat{I}}^1$.*

Proof. During the message exchanges 3 cases might happen. The first case is when one of the nodes has packets from flow f_i, and the other node is empty or has packets from the same flow. The message exchange in this case is similar to the message exchange in the case of single flow. When the two nodes carry two different flows, the sender performs transmission similarly to the intra-flow method, but the receiver mixes the received packets with the packets in its buffer. Assume that the receiver later meets the destinations. The destinations can easily remove the packets that belong to them, and construct intra-coded packets of the opposing flow. Thus, inter-flow NC does not have any negative impact on the delay in this case. In the last case, the sender has joint inter- and intra-flow coded packets. Any transmission of the form f_1+f_2 is similar to two separate transmissions f_1 and f_2. The reason is that, similar to the previous case, each joint inter- and intra-flow coded packet can be easily reduced to intra-flow coded packets by the destinations. ∎

A direct result that can be inferred from Theorem 2 is that inter-flow NC removes competition between the flows for using the buffer and bandwidth resources.

2) General Case: Assuming that the buffer size and bandwidth of the nodes are equal to 1 packet, the spread speed of the packets in the network can be modeled as a disease epidemic. If we define the nodes with nonempty and empty buffers as the *infected* and *susceptible* nodes, then the rate at which the nodes become infected is $dx/dt = \lambda x(1-x)$. Here, x and λ are the fraction of infected nodes and the average number of contacts of any given node. Multiplying x and $(1-x)$ gives the probability of contacts between infected and susceptible nodes. Solving the equation, the fraction of infected nodes at time-slot t can be calculated as [14]:

$$x(t) = \frac{x_0 e^{\lambda t}}{1 - x_0 + x_0 e^{\lambda t}} \tag{5}$$

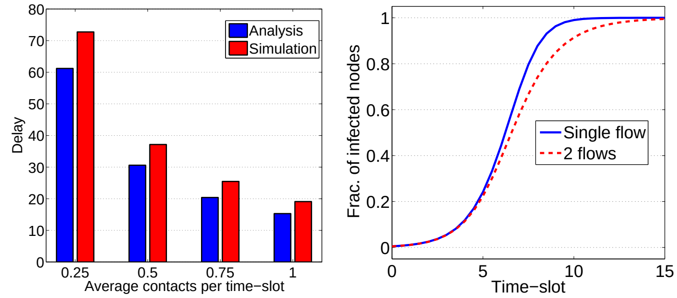
here, x_0 represents the fraction of initial infected nodes, which in our case is equal to $1/n$. Much similar to the work in [15], we assume that each node can transmit an innovative packet to its neighboring node. The authors in [16] show that, in the case of abundant buffers, the probability that a transmitted coded packet is innovative to its neighbor is $1-1/q$, where q is the size of a finite field (Galois field). A typical used field size is $q = 2^8$; therefore, $1-1/q$ is sufficiently close to 1. In our analysis we do not assume abundant buffers; however, we show that our analysis is close to the simulation results.

Theorem 3. *In the case of single flow, and buffer size, and bandwidth equal to B packets, the delay of intra-flow NC is:*

$$D_I = \frac{1}{\lambda} \ln \left[e^{\ln(n) + \frac{m}{B}} - n + 1 \right] \tag{6}$$

Proof. Following Equation (5), the rate at which the destination node meets the infected nodes is equal to $\lambda x(t)$. The bandwidth and buffer sizes are equal to B, so the destination receives coded packets with a rate equal to $\lambda B x(t)$. Consequently, the number of received packets from time-slot 0 to D_I can be calculated as follows:

$$\int_0^{D_I} \lambda B \frac{\frac{1}{n} e^{\lambda t}}{1 - \frac{1}{n} + \frac{1}{n} e^{\lambda t}} = B \ln\left(n + e^{\lambda t} - 1\right)\Big|_0^{D_I}$$



(a) (b)

Fig. 5. Number of nodes $n = 200$ and bandwidth $b = 1$. (a) Comparing analysis and simulation result. Buffer size $B = 1$. (b) The fraction of infected nodes. Buffer size $B = 1$ per flow and a shared bandwidth $b = 1$.

which is equal to the number of received packet, m. As a result, we have $B[\ln(n + e^{\lambda D_I} - 1) - \ln(n)] = m$. Solving this equation results in Equation (6). ∎

We verify our delay analysis by comparing it with simulation results. We set the number of nodes and number of packets to 200 and 10, respectively. Moreover, the buffer size and bandwidth of the nodes are equal to 1. We run the simulations 100 times and take the average. Fig. 5(a) shows the comparison between the analysis and simulation for different average numbers of node contacts. As the figure shows, there is a gap between analysis and simulation, which is due to the linearly dependence of some received packets by the destination node. When joint inter- and intra-flow NC is used, there is no competition between the flows. Consequently, the delay of joint inter- and intra-flow NC will be the same as single flow transmission (Equation (6)). We leave the rest of verifications to the Section V.

Consider the case of only intra-flow NC, and the existence of 2 flows. If we assign half of the bandwidth and buffer resources to each flow, the propagation of the 2 flows become independent, and we can directly use Equation (6) to compute the delay of each flow for a buffer size and bandwidth equal to $B/2$. However, with a shared bandwidth equal to 1 packet, we cannot assign a separate bandwidth to each flow, and Equation (6) cannot be used. Thus, we fist compute the fraction of infected nodes, and then use the result to calculate the delay of the method.

Consider flows f_1 and f_2 between nodes s_1-s_2 and s_2-s_1, respectively. Assume that the buffer size of the relay nodes is equal to 2, and each relay node assigns one location of the buffer to each flow. Also, the bandwidth of the nodes is equal to 1 packet. Assume that the relays r_i and r_j meet each other, and packets a and b belong to flows f_1 and f_2, respectively. The events that result in packet transmission between the nodes are shown in Figs. 6(a) to (d). If node r_i has packet a or b and the other node is empty, node r_j will be infected. The case that node r_i has packet b in its buffer has not been shown, as that is similar to Fig. 6(a). If at least one of the nodes does not have the packet of the other node, both of them become infected by both of the flows, as shown in Figs. 6(b) and (c). However, when node r_i is infected by both of the flows and

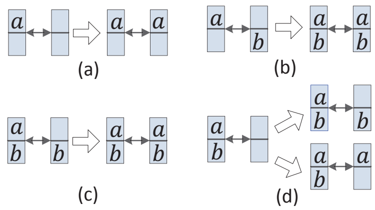


Fig. 6. The possible cases for infection propagation between the nodes.

the other infected by none of them, r_i can transmit just one of the packets randomly, as shown in Fig. 6(d).

Assume that x, y, and z represent the fraction of infected nodes by flows f_1, f_2, and both of them, respectively. The probability that an infected node by f_1 meets a susceptible node by f_1 is $\lambda(1-x)x$. However, if the nodes that belong to z meet susceptible nodes, the susceptible nodes' probability of infection will be equal to 0.5. As the flows are independent and their spearing rate can be assumed to be the same, we have $z = x \times y = x^2$. Consequently, the spreading rate of flow f_1 is equal to $dx/dt = \lambda(1-x)(x-x^2/2)$. As a result, we have $2dx/x(x-1)(x-2) = \lambda dt$. By taking integral of the left and right hand sides respect to x and t, respectively, we have $\ln[x(x-2)/(x-1)^2] = \lambda t + c$. Raising base e to the power of the both sides and defining $C = e^c$ gives $x(x-2)/(x-1)^2 = Ce^{\lambda t}$, with solution:

$$x(t) = \frac{Ce^{\lambda t} + \sqrt{1 - Ce^{\lambda t}} - 1}{Ce^{\lambda t} - 1} \quad (7)$$

and assuming that x_0 is the fraction of infected nodes at time-slot 0, C can be calculated as $C = \frac{x_0(x_0-2)}{(x_0-1)^2}$.

Theorem 4. *Consider 2 flows that are assigned separate buffers equal to 1 packet. Also, assume that the shared bandwidth between the flows is equal to 1 packet. The delay of intra-flow NC is equal to:*

$$D_I^2 = \frac{1}{\lambda} \ln\left(\frac{Ce^m - 2e^m + 2}{C}\right) \quad (8)$$

We leave the proof of this theorem in the Appendix.

C. Spray-and-Wait with Joint Inter- and Intra-flow NC

In order to limit the number of transmissions and the number of packet copies in the network, we can combine joint inter- and intra-flow NC with the spray-and-wait routing method [13]. In the spray and wait method, the source node assigns a number of tokens to each packet. Once a node transmit a packet to its neighbor, it gives half of the tokens to the neighbor. The packets whose tokens are not larger than 1 cannot be transmitted to the other relay node; however, they can be directly forwarded to the destination node when a contact happens.

When we use NC, we cannot assign the tokens to the original packet, so we assign t tokens to the whole set of packets of a flow. When two non-destination node meets each other, they first redistribute their tokens in proportion to their ranks. Assume that the two encountered nodes r_i and r_j has t_i and t_j tokens. Moreover, the coefficient matrices of the node r_i and its neighbor r_j are c_i and c_j, respectively. Node

Algorithm 4 The SWIINC algorithm (source s_i side)

/*Initialization*/
Set $t_i = t$
/*On encountering node r_j*/
Run Algorithm 6
/*On receiving packets*/
Store the received packets in the buffer

Algorithm 5 The SWIINC algorithm (relay r_i side)

/*Initialization*/
Set $t_i = 0$
/*On encountering node r_j*/
Run Algorithm 6
/*On receiving packets and t tokens*/
if buffer is full **then**
  Mix the packet with the buffered packets
else Store the packet in an empty location

Algorithm 6 On $i$the node encountering the jth node

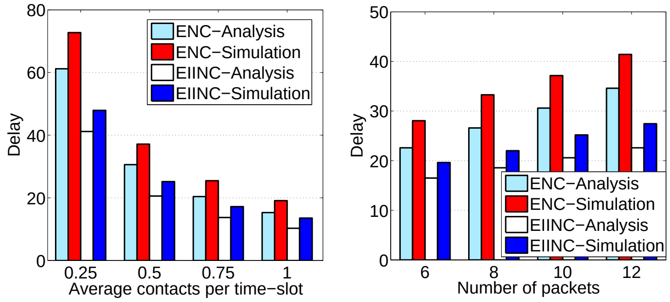
Transmit the coefficients matrix, c_i
Receive the coefficients matrix of the neighbor, c_j
$g_i = rank(c_i \cup c_j) - rank(c_j)$
Redistribute the tokens with r_j
while $g_i > 0$ and $t_i > 1$ **do**
  Send a random linear coded packet
  $t_i = t_i - 1$, Redistribute the tokens with r_j

r_i will have $Round[(t_i + t_j)c_i/(c_i + c_j)]$ tokens after the redistribution of the tokens. Here, Round is the closest integer to the result. Then, each of the two nodes transmits a random linear coded packet to the other node if it has innovative packets for the neighbor, and reduces its tokens by one. The two nodes repeat the redistribution and transmission phases until their bandwidth allows that. The processes at the source and relay nodes are shown in Algorithms 4 and 5, respectively.

V. Simulations

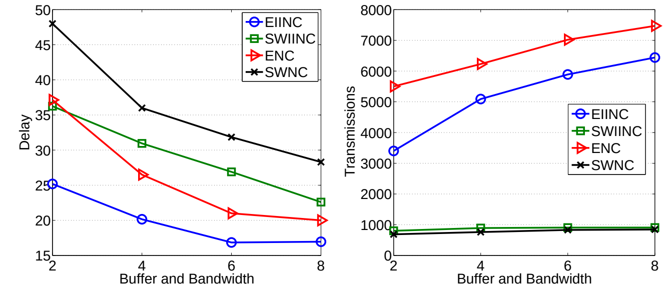
In this section, we evaluate our epidemic routing with inter- and intra-flow NC (EIINC) and spray-and-wait with NC (SWIINC) methods, with the schemes in [3]. We refer to the proposed epidemic and spray-and-wait NC methods in [3] as ENC and SWNC, respectively. In the ENC method, the sources transmit intra-flow coded packets of the stored packets in their buffer to their neighbors. Then, the relay nodes continue this process until the destination node receives m linearly independent packets. The SWNC is similar to the ENC method, but the m packets of each flow have a given number of tokens. The source nodes share the tokens with their neighbors, proportional to the rank of packets in their buffer. In order to evaluate the methods, we implemented a simulator in the MATLAB environment.

We evaluate the methods on both synthetic data and real data trace. For the synthetic evaluation, we assume that each node meets any other nodes with a fixed probability, and the average



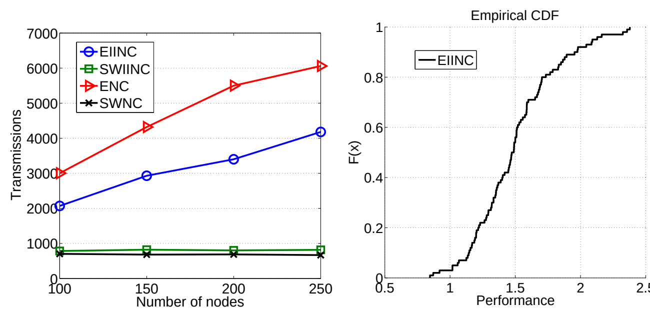
(a) (b)

Fig. 7. Comparing the delivery delay in terms of time slot, synthetic data, $B = 2$, $b = 2$, and $n = 200$. (a) $m = 10$. (b) $\lambda = 0.5$.



(a) (b)

Fig. 8. Synthetic data, $m = 10$, $n = 200$, $\lambda = 0.5$, $t = 500$ per flow. (a) Delivery delay in terms of time slot. (b) Total number of transmissions.



(a) (b)

Fig. 9. Synthetic data, $m = 10$, $B = 2$, $b = 2$, $\lambda = 0.5$, $t = 500$. (a) Total number of transmissions. (b) The Empirical CDF of the EIINC method' performance over the ENC method, $n = 200$.

number of contacts of the different nodes are the same. We also simulate the methods on the INFOCOM 2006 trace [17]. We run the simulations 100 times with random source nodes, and for the plots, we use the average outputs. For EIINC and SWIINC methods, which use joint inter- and intra-flow NC, the whole buffer of the nodes is assigned to all of the flows. However, for the ENC and SWNC methods that only use intra-flow NC, we assign separate parts of the buffers, with equal capacity to each flow. Also, half of the bandwidth is assigned to each flow in the ENC and SWNC methods.

1) Synthetic Data: We consider a network with 200 nodes. There are two source nodes; each of them wants to transmit 10 packets to the other source node. Moreover, the buffer size and bandwidth of the nodes are equal to 2 packets. As Fig. 7(a) shows, the delay of the EIINC method is less than that of the ENC method, which is due to the use of joint inter- and intra-flow NC. Furthermore, the delay of both methods decreases as we increase the average number of contacts of the nodes. The figure shows that there is a gape between our analysis and the simulation results. The reason is that in the analysis we assumed that each received packet to the destination is useful; however, in reality the destination nodes receive some linearly dependent packets. It can be inferred from the figure that the delay of ENC method is up to 50% more than the proposed EIINC method method.

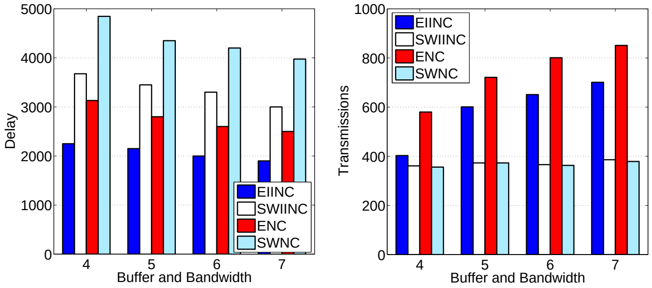
In the next experiment, we evaluate the effect of packet size on the delay. As Fig. 7(b) depicts, the delay of the ENC and EIINC methods increase as we increase m. The reason is that when more packets are coded together, the destination nodes needs to receive more packets to be able to decode and retrieve the original packets. Moreover, the efficiency of the EIINC method over the ENC scheme increases as we increase the packet size.

We next study the effect of buffer size and bandwidth on the delivery delay. We set the bandwidth equal to the buffer size, and change it from 2 to 8. Moreover, the number of nodes and λ are equal to 200 and 0.5, respectively. We assign 500 tokens to each flow. As Fig. 8(a) shows, the delay decreases as we increase the buffer size, which is due to transmission of more packets in each contact of the nodes. By increasing the bandwidth and buffer size, some of the bottlenecks in the ENC methods are solved. That is why the gap between the EIINC and ENC methods decreases as we increase the bandwidth and buffer size. The number of tokens in the SWIINC and SWNC methods are fixed, but the propagation speed of the packets increases as we increase the buffer size and bandwidth. That is why a larger buffer size and bandwidth decreases the delay of the SWIINC and SWNC methods.

Fig. 8(b) shows the effect of bandwidth and buffer size on the number of transmissions. The settings are the same as those in Fig. 8(a). As expected, the number of transmissions in the EIINC and ENC methods increases as we increase the bandwidth and buffer size. However, the number of transmission in the SWIINC and SWNC methods are almost fixed, which is due to the limited number of tokens of each flow.

We change the number of nodes in the range of 100 and 250, and show the number of transmissions in Fig. 9(a). The bandwidth and buffer size are equal to 2 packets. Moreover, we fix λ to 0.5 contacts per times-lot. The figure depicts that the number of transmissions in the EIINC and ENC methods increases in the more dense networks, which is due to number of contacts. However, the number of transmissions in the SWIINC and SWNC methods are almost fixed.

We divide the delay of the ENC method by that of the EIINC method, and plot the CDF function in Fig. 9(b). The bandwidth and buffer size are set to 2. Moreover, λ and n are equal to 0.5 and 200, respectively. The figure depicts that in only 3% of the cases the delay of the ENC method is less than that of the EIINC method, which is due to the randomness of the contacts. In about 30% of the runs, the delay of the ENC



(a) (b)

Fig. 10. Comparing the delay, in terms of seconds, and number of transmissions of the methods on INFOCOM trace, $t = 200$ per flow.

method is between 1.5 and 2 times that of the EIINC method. In addition, in 60% of the cases, the delay of the ENC method is up to 1.5 times that of the EIINC scheme.

2) Real Data Trace: In the first experiment on real trace, we evaluate the effect of buffer size and bandwidth on the delay. As Fig. 10(a) shows, the proposed EIINC method provides the lowest delay compared to the other approaches. The SWIINC and SWNC methods, which have limited number of copies of the packets, results in more delay than the EIINC and ENC methods. As we increase the buffer size, more packets can be carried by the nodes, which reduces the delay of the EIINC and ENC methods. Also, more bandwidth and buffer size increases the spreading speed of the packets, which decreases the delay of the SWIINC and SWNC approaches.

Fig. 10(b) shows the number of transmissions of the methods. As expected, the number of transmission in the SWIINC and SWNC methods does not change as we increase the buffer size and bandwidth. The number of transmissions in the EIINC and ENC methods increases as we increase the buffer size and bandwidth, which is is because of more available resources.

VI. Conclusion

Because of the nondeterministic mobility of nodes, and the lack of continuous network connectivity, data transmission in DTNs is challenging. In order to increase the efficiency of the data transmission in DTNs, some previous works combine intra-flow NC with the routing protocols. In this paper, we propose two routing mechanisms using joint inter- and intra-flow NC for the purpose of data exchange between the source nodes in DTNs. We discuss two reasons why inter-flow NC helps to reduce the delivery delay of the packets, and we show the benefit of the joint inter- and intra-flow NC, compared with just intra-flow NC both analytically and empirically. Our simulation results show that joint coding can reduce the delay up to 40% when compared to just intra-flow coding.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003, pp. 27–34.

[2] J. Widmer and J. L. Boudec, "Network coding for efficient communication in extreme networks," in *ACM WDTN*, 2005, pp. 284–291.

[3] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang, "Benefits of network coding for unicast application in disruption-tolerant networks," *IEEE/ACM Transactions on Networking*, 2012.

[4] H. Chun, N. Ning, and F. Gang, "Network coding based routing scheme for resource constrained delay tolerate networks," in *ICCP*, 2012, pp. 40–45.

[5] S. Ahmed and S. S. Kanhere, "HUBCODE: hub-based forwarding using network coding in delay tolerant networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 19, pp. 828–846, 2011.

[6] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[7] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[8] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[9] P. Ostovari, J. Wu, and A. Khreishah, "Network coding techniques for wireless and sensor networks," in *The Art of Wireless Sensor Networks*, H. M. Ammari, Ed. Springer, 2013.

[10] P. Ostovari and J. Wu, "Towards network coding for cyber-physical systems: Security challenges and applications," in *Security and Privacy in Cyber-Physical Systems: Foundations and Applications*, H. Song, Ed. Wiley, 2017.

[11] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "On the benefits of random linear coding for unicast applications in disruption tolerant networks," in *IEEE WiOpt*, 2006, pp. 1–7.

[12] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., 2000.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *ACM WDTN*, 2005, pp. 252–259.

[14] M. Newman, *Networks: An Introduction*. Oxford University Press, 2009.

[15] Y. Lin, B. Li, and B. Liang, "Stochastic analysis of network coding in epidemic routing," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 794–808, 2008.

[16] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: a network coding approach to optimal multiple rumor mongering," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2486–2507, 2006.

[17] J. Scott et al, "CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006, May 2009.

Appendix

Proof of Theorem 4: The rate at which the destination node receives the packets is equal to $x(t)\lambda$. Consequently, from Equation (7) we have:

$$\int_0^{D_I^2} \lambda \frac{Ce^{\lambda t} + \sqrt{1 - Ce^{\lambda t}} - 1}{Ce^{\lambda t} - 1} = m$$

which gives us:

$$\lambda t + 2arctanh(\sqrt{1 - Ce^{\lambda t}})\Big|_0^{D_I^2} = m$$

and

$$\lambda D_I^2 + 2arctanh\sqrt{1 - Ce^{\lambda D_I^2}} - 2arctanh\sqrt{1 - C} - m = 0$$

By replacing $arctanh(z)$ with $\frac{1}{2}\ln\frac{1-z}{1+z}$ we can rewrite the above equation as:

$$\lambda D_I^2 + \ln\frac{2 - Ce^{\lambda D_I^2}}{Ce^{\lambda D_I^2}} + \ln\frac{2 - C}{C} - m = 0$$

thus, $\ln[2 - Ce^{\lambda D_I^2}/(2 - C)e^{\lambda D_I^2}] = m - \lambda D_I^2$. Raising base e to the power of the equation gives us: $\frac{2 - Ce^{\lambda D_I^2}}{(2-C)e^{\lambda D_I^2}} = e^{m - \lambda D_I^2}$ with solution $D_I^2 = \frac{1}{\lambda}\ln(\frac{Ce^m - 2e^m + 2}{C})$.