A transmission protocol based on network coding in many-to-one delay tolerant networks

Jang-Ping Sheu*, Chih-Yin Lee and Chuang Ma

Department of Computer Science,
National Tsing Hua University,
Hsinchu 30013, Taiwan
Email: sheujp@cs.nthu.edu.tw
Email: s9962531@m99.nthu.edu.tw
Email: machuang@mx.nthu.edu.tw
*Corresponding author

Abstract: In delay tolerant networks (DTNs), the delay of packets is long due to the intermittent connections caused by the mobility of nodes. Epidemic routing protocol (ERP) can reduce the delay time and raise the packet delivery rate by replicating and spread copies of source packets in the networks, but it inevitably exhaust memory resources and network bandwidth. In this paper, we propose a network coding with limited buffer routing protocol based on two network models, single packet network model and multiple packets network model. When a buffer of a node is full, two packets in the buffer are chosen randomly and encoded linearly together into a packet to save buffer space and improve the performance in many-to-one communications. Moreover, in the transmitting stage of our network coding routing protocol, we proposed two efficient approaches that can enhance the efficiency of information exchange and packets transmission in communications of delay tolerant networks.

Keywords: DTNs; delay tolerant networks; epidemic routing; limited buffer; multicast; network coding.

Reference to this paper should be made as follows: Sheu, J-P., Lee, C-Y. and Ma, C. (2015) 'A transmission protocol based on network coding in many-to-one delay tolerant networks', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 19, Nos. 1/2, pp.19–28.

Biographical notes: Jang-Ping Sheu received the BS in Computer Science from Tamkang University, Taiwan, Republic of China, in 1981, and the MS and PhD in Computer Science from National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively. He is currently a Chair Professor of the Department of Computer Science, National Tsing Hua University. His current research interests include wireless sensor networks, cognitive networks and cloud computing. He is an IEEE Fellow.

Chih-Yin Lee received the Master degree in Computer Science from National Tsing-Hua University, Taiwan, in 2012. Her research interests include wireless sensor networks, delay tolerant networks and network coding.

Chuang Ma received the BS in Automatic Control from Harbin University of Science and Technology, China, in 1997, the MS in Software Engineering from Harbin Institute of Technology, China, in 2004, and PhD in Computer Architecture from Harbin Institute of Technology, China, in 2011, respectively. He is postdoctoral fellow of the Computer and Communication Research Center, National Tsing Hua University, Taiwan, Republic of China. His current research interests include wireless communications and mobile computing. He is an IEEE Member.

This paper is a revised and expanded version of a paper entitled 'An efficient transmission protocol based on network coding in delay tolerant networks' presented at *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2013)*, Taichung, Taiwan, 3–5 July, 2013.

1 Introduction

Nowadays, delay tolerant networks (DTNs) (Fall, 2003) are applied in many areas of surveillances, such as medical treatment, geologic structure and living beings activities. However, there are some challenges in these applications. For example, the communications among sensor nodes are often disrupted with low communication bandwidth, such characteristics of networks will lead to low throughput and long end-to-end delay of packets. Solutions have been proposed to handle routing and make improvements in the research area, where connections between wireless mobile nodes are intermittent over time due to small transmission range, mobility or interferences. Moreover, the end-to-end path between source and destination does not exist due to the disrupted connections, especially in the network model with mobile nodes, only when a pair of nodes moves into the communication ranges of each other, the temporal link is established. So the path that a packet goes from source to destination could be seemed as an opportunistic path, which means every link in the path appears by chance. While traditional routing approaches relying on well-connected end-to-end routes cannot be applied to DTNs.

In addition, the size of buffer in nodes is also closely related to the performance of the routing protocols in DTNs. Unfortunately, the size of buffer are limited in DTNs applications because sensor nodes are usually designed with low cost purpose, and consequently with limited capacity of memory. Because of the intermittent nature of DTNs, nodes adopt carry-and-forward method, where a node would keep a packet in its buffer for a long time, and forward it many times to increase the delivery ratio of the packet, so small buffer size deteriorate the performance in DTNs greatly. Also, when buffer size is limited, the insufficiency of utilisation of each buffer space arises. If amounts of copies of each source packet are uneven, then there must be some source packets have little copies spread in the network, and it will worsen the coupon collector effect (Motwani and Raghavan, 2010) on destination, which means more source packets are being collected and the probability to get innovative packets from each meeting on the destination becomes smaller, so it will take a long time to collect all innovative packets.

Thus, how to evenly spread the copies of each source packet is a very important issue in many-to-one network model, many source nodes and one destination node, with limited buffer. Epidemic routing protocol (ERP) (Vahdat and Becker, 2000) has been proposed to deal with the disrupted connection in DTNs. It is a flooding-based routing protocol, and its behaviour is analogous to the spread of infectious disease. Nodes in ERP continuously replicate and transmit packets to newly contact nodes that have not possessed a copy of the packets. However, in researches area of DTN, the trade-off between delivery rate and cost of resource is focused on now. ERP can raise the delivery rate and shorten the end-to-end delay of the packets, but cost lots of resources during the flooding, especially when buffer size is limited. Also, ERP is not effective when the opportunistic encounters between nodes are not purely random. Although ERP has greatly improved the performance of transmitting packets in DTNs, the problems of ERP described above become severe especially when buffer size is limited.

In this paper, we introduce network coding method in network layer to solve the problem brought by limited buffer size to raise the packet delivery rate and shorten the packet delay. With network coding, a fewer transmissions are needed than the traditional store-and-forward mechanism. We proposed an efficient transmission protocol, network coding protocol for limited buffer size (NC-LB), based on network coding for many-to-one model in DTNs to deal with limited buffer constraint. Instead of applying network coding only to the packets that come from the same source nodes, such as in the unicast-based models, we apply the network coding to the packets from multiple sources which are generated in the same period of time. Traditional gain brought by network coding technique is to reduce the number of packet transmissions. However, the main purposes in this paper include increasing the packet delivery rate and decreasing the packet delay which are the two most important performance metrics in DTNs. The advantage of our NC-LB is to increase the packet delivery rate when the buffer size is limited. Nodes under our protocol would encode the incoming packet with a packet in the buffer, that the coded packet would keep the partial information of both packets. In comparison, when buffer is full, nodes under traditional technique must either drop the incoming packet or one packet from its buffer, thus good dropping strategies are important to decide that packet should be dropped to let copies of every source packets evenly spread in the network. In our NC-LB of multiple-packet network model, we apply two kinds of packets dropping strategies LD and PWD for each generation of packets in local buffer and the receiver's buffer to decide how many packets of each generation should be sent. Our analysis and simulation shows that NC-LB delivers packets with shorter delay than ERP. With more source nodes, NC-LB can save more than 70% delivery delay in many-to-one model than ERP. On the other hand, it also is shown that with the same delays required, NC-LB requires less buffering capacities than ERP.

The rest of this paper is organised as follows. We discuss related work of DTNs protocols and network coding-based protocols in Section 2. In Section 3, we propose NC-LB protocol in many-to-one model with buffer size constraint. The simulation results are shown in Section 4. Section 5 concludes our research.

2 Related work

Various routing protocols were proposed in DTNs with mobile nodes. The basic assumption of these DTNs is low density of nodes and the approximate random meeting probability for each two nodes. Moreover, raising the packet delivery rate and shortening the packet delay are the two most important performance metrics in DTNs.

There are two kinds of protocols well known in the studies of DTNs, including direct transmission mechanism and replicate-and-forward mechanism. The early idea proposed is direct transmission protocol. In direct transmission protocol, a source node transmits packets to destination node only when they are directly connected with each other. Without any help of relaying, however, direct transmission protocol suffers from longer delay time. Considering the drawback of direct transmission, replicate-and-forward mechanism, such as ERP, was proposed. To reduce the delay time from source to destination, ERP involves relay nodes to help forwarding packets to destination. With the help of relay nodes, end-to-end delay is efficiently reduced. However, the overhead of redundant copies is produced as the cost, because relay nodes of ERP will keep all the packets it received until the packets are expired. Although the more existence of copies in network increases the probability of reaching destination for packets, most copies become redundant once the destination gets one of the packets, and the redundant copies may occupy the memory resources rapidly. Therefore, it is a major issue in the study of routing in DTNs to balance the overheads of memory consumption and end-to-end delay time.

Studies about ERP are proposed to solve this trade-off problem based on forwarding strategies, including restricted ERP (Abdulla and Simon, 2008), history-based (Ferrari Aggradi et al., 2008) and social-based (Hossmann et al., 2010; Li et al., 2010). Serving as a typical protocol, restricted ERPs are proposed to reduce redundant copies in epidemic routing. In restricted ERPs, a relay node decides whether to forward or delete a packet in its buffer by considering the packet's expiration time, time to live. In Lu and Hui (2010), the *n*-epidemic protocol was proposed to reduce the number of transmissions, but the delay time of epidemic routing would not be increased. In *n*-ERP, nodes would only forward packets when they meet more than two nodes at the same time, and each packets in relays will be forwarded for at most *n* times, so the energy efficiency is also improved by reducing the amount of transmissions. In addition, buffer size is a bottleneck of packets storage and broadcast in the application of ERP. A buffer is used to temporarily hold data while it moves from one place to another place, so the size of a buffer determines the capability of holding data. When buffer is full, nodes under ERP must either drop the incoming packet or one packet in its buffer, thus ERP would need a good dropping strategies (Fathima and Wahidabanu, 2011) to decide which packet should be dropped to let copies of every source packet evenly spread in the network. In a word, nodes need dropping strategy to decide which packets are to be dropped when their buffers are full. Except employing forwarding strategies to balance the trade-off between memory consumption and packet delay time, the dropping strategies in ERP are addressed in Lindgren and Phanse (2006). A dropping strategy can enormously affect the distribution of copies of each source packet. Many mechanisms had been proposed to balance the distribution of copies of each source packet, such as Drop Old, Drop Random, FIFO, and evict MOst FOrwarded first (MOFO) (Lindgren and Phanse, 2006).

Network coding (Ahlswede et al., 2000) is a popular technique in recent researches. With network coding, fewer transmissions are needed than the traditional store and forward mechanism. The benefit of network coding mechanism in DTNs is less buffer size requirements, as a result, network coding has also been applied to unicast and multicast. The effect of applying network coding mechanism to the unicast was studied in Lin et al. (2008), Li et al. (2012), and Yoon and Haas (2010), but these studies are under the assumption that nodes can store only one single packet, or bandwidth is only enough to transmit one packet in each meeting. And an approach of applying random network coding to multicast in wireless sensor networks was presented in Jin et al. (2011). In Yunfeng et al. (2008), the authors proposed a network coding algorithm based on ERP, NCER, which is different with original network coding. In NCER, network coding is applied in both the source and the destination. The author assumed that a source has a large number of packets to be transmitted to a destination, so arriving packets are a stream in the queue. Moreover, in this algorithm, this stream in the queue is partitioned into blocks and each block are linearly combined by *k* packets as a pseudo packet with a set of coefficient which is chosen from Galois Field (Lidl et al., 1997). The practical way of implementing network coding was proposed in Chou et al. (2003), which proposed a packet format and buffering model, and simulated such a practical network coding system in several network topologies. The authors showed that in a network without any buffer management strategy but with random network coding technique, when two nodes meet, the probability of providing each other with innovative packets is high. Also the destination node gets innovative packets from every encountering with high probability. On the contrary, destination node in ERP would suffer from coupon collector effect. In Qin and Feng (2013), the authors focused on the performance modelling and evaluation for random linear network coding (RLNC)-based epidemic routing in DTNs with limited transmission capacity. Considering that multiple unicast communications co-exist and competitions for the transmission capacity in the network, the authors developed an analytical model to trace the transmission process and evaluate the transmission performance of the RLNC-based routing protocols. In Zeng et al. (2013), the authors proposed a dynamic segmented scheme based on network coding to efficiently exploit the transmission opportunity in DTNs. By the segmented network coding mechanism, they made the transmission adapt to the dynamics of the network and derived a lower bound of the expected delivery delay for bulk-data dissemination. Also, the authors used both analytical method and simulation results to validate the high performance of their scheme.

3 Protocol design

In this section, we focus on the problem how to evenly spread the copies of each source packet in many-to-one network model with limited buffer size. On the basis of many-to-one model, our network model is described as follow. There are *S* source nodes, *R* relay nodes and one destination node. Every node has the ability to move in scenario, only source nodes can generate new packets by sensing devices. And we assume that the data generating rates are same on all the source nodes. Each source node and the number of buffer size of each relay node is *BS*, where can only contain *BS* packets at a certain time. Moreover, we assume that the destination node is equipped with enough resource, such as buffer space and memory, to contain all received packets. In other words, the destination node will not drop any packet which it has received. Furthermore, we assume that nodes can randomly move in a constrained area, and the area is large enough comparing to the transmission range, so that the probabilities of nodes meeting each other is low in accordance with the features of the DTNs. We also let nodes periodically transmit and listen low power beacon messages. When two nodes hear beacon message from each other, it means they are within an available mutual transmitting range. In our network model, it is not necessary to constrain the bandwidth between nodes because the limited bandwidth constraint is similar to the limited buffer constraint. For example, if nodes can only transmit at most four packets during each encounter, then it is analogous to our network model where every node has a buffer which is limited to four.

3.1 Single-packet network model

We will propose our protocol based on ERP in single-packet network model at first, which is the special case of the multiple-packet model. In single-packet model, each source node only generates one packet at the beginning. In the following, we will provide details of applying ERP onto many-to-one model in DTNs with limited buffer constraint.

3.1.1 ERP for single-packet network model

Each node in ERP stores and maintains a summary vector (SV) that is compact representation information of all packets stored in its buffer. When two nodes meet, they will exchange their summary vectors and decide which packets will be transmitted based on their summary vectors. For example, when two nodes A and B come into communication range of each other, they first exchange their summary vectors. Without loss of generality, we assume B will transmit packets to A first. B decides the packets which will be transmitted to the A by performing a logical negation to SVA, and then performing AND operation between SVA and SVB to determine which packets have not been received by A. (SVA AND SVB) represents the set of packets in B and not in A. Then B would start to transfer packets (SVA AND SVB). In the receiving stage, a node would receive any packet if its buffer is not full, otherwise, it would start to drop packets for receiving novel packets. For example, when node A receives a packet, if its buffer is not full, the packet would be put into the buffer directly; but if the buffer is full, it needs dropping strategy to decide whether it should drop the incoming packet, or drop an old packet in its buffer to receive the incoming one. When the amount of copies of a certain source packet is few, the probability that the destination can receive the packet would be low, and a good dropping strategy can help to reduce the number of copies of source packets. In other words, a good dropping strategy makes the amounts of copies of each source packets to be approximate equality in the network, so the destination node has the same probability to acquire every source packet.

We apply MOFO dropping strategy to ERP. MOFO gives the best performance comparing to all the other dropping strategies. In MOFO, nodes would keep a times record of a packet being transferred. When two nodes meet, a node would receive new packets it does not have from the other node. If a node's buffer is full, it would sort and choose the packet in its buffer that has been transmitted the most of times as the candidate packet to be dropped in this round. After that, the node will check whether the new packet has been received and dropped before. If it is not, the candidate packet will be dropped. If it is, it would compare the transmission times of the new packet with the candidate packet in the buffer, if the new packet has been transmitted more, it would not be received; otherwise, the candidate packet would be dropped, and the new packet would be received.

3.1.2 NC-LB for single-packet network model

Our proposed protocol, network coding based on limited buffer (NC-LB), adopt the idea of carry-and-forward as epidemic routing. But for the reason of limited buffer, we apply the network coding technique to the packets in the buffer both in receiving stage and in transmitting stage to enlarge the information amount of receiving.

We first define S as the number of source nodes, and 'source packet' as the packet has not been encoded, 'coded packets' as the packet, which is encoded by the linear combination of source packets. We also define P_i as a source packet or its copy which is generated by the source node i. A linear coded packet x which is the linear combination of $P_1, P_2, \ldots, P_S$ can be denoted in the form: $x = \sum_{i=1}^{S} \alpha_i P_i$, where α_i are encoding coefficients and are randomly chosen from Galois Field. Note that, a source packet P_k can also be denoted in the form: $x = \sum_{i=1}^{S} \alpha_i P_i$, with the coefficients α_i ($i \neq k$) is zero, only α_k is non-zero. A coded packet that is consisted of P_{j1}, P_{j2}, … and, P_{jm} can also be denoted in the form: $x = \sum_{i=1}^{S} \alpha_i P_i$, with the coefficients α_i ($i \neq j_1, j_2, \ldots, j_m$) are zeros and $\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jm}$ are non-zero. Each packet x in the buffer stores the encoding vector $[\alpha_1, \alpha_2, \ldots, \alpha_S]$ of the packet.

3.1.2.1 Receiving stage

In NC-LB, every node does not drop any packet even when its buffer is full, but linearly encode the incoming packet with a random packet stored in its buffer. We assume node A and node B denote the two meeting nodes, and A starts the receiving process. If A's buffer is not full, and A receives a packet from B, A would simply put the packet into one empty position in buffer. Otherwise, if A's buffer is full and A receives a packet $x_b = \sum_{i=1}^{S} \beta_i P_i$ from B, A will randomly pick a packet $x_a = \sum_{i=1}^{S} \alpha_i P_i$ from its buffer as candidate of encoding, randomly generate two coefficients γ_1, and γ_2 from Galois Field and combine x_a and x_b. The new linearly combined packet is $x' = \gamma_1 \times x_a + \gamma_2 \times x_b = \gamma_1 \times \sum_{i=1}^{S} \alpha_i P_i + \gamma_2 \times \sum_{i=1}^{S} \beta_i P_i = \sum_{i=1}^{S} (\gamma_1 \alpha_i + \gamma_2 \beta_i) \times P_i$, which is a new linear combination of source packets, then A would replace x_a with x' in its buffer space.

A destination node will obtain either a source packet or a coded packet when it meets a mobile source node, and attempts to decode the source packets from the coded packets. Decoding S source packets from S coded packets is equivalent to solving the S linear independent equations of S unknown variables. The *decoding matrix* represents the coefficient matrix of such linear equations. When the rank of the *decoding matrix* is *S, Gaussian Elimination* can be applied to solve this problem and the S source packets can be decoded from the encoded packets. However, if the destination node has received S packets but rank of the *decoding matrix* is $<S$, then the destination node has to wait to receive more packets and linearly encode the new received packets to the original S packets, until the rank reaches S.

3.1.2.2 Transmitting stage

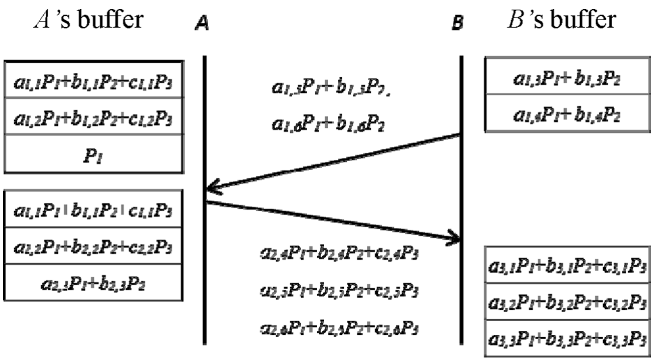
In transmitting stage, it is unnecessary for encountering nodes to always transmit all buffered packets to each other, so that we proposed two approaches to decide how many packets will be transferred when two nodes meet. According to the local packets information in the buffer, we proposed the first approach, *local-decision* (LD), to decide the number of packets to be transferred. The second approach is *Pair-Wisely-Deciding* (PWD), it considers not only the local buffer information, but also receiving node's buffer information.

We define $X_i = \{x \mid x$ is a source packet or an encoded packet in node i's buffer $\}$, and $|X_i|$ being the number of total packets in node i's buffer. In LD approach, nodes need to keep a record of the packets in their buffers. For node i, we define $\text{Record}_i = \{k \mid \text{for all packet } x \in X_i,\ x = \sum_{k=0}^{S} \alpha_k P_k$ and $\alpha_k \neq 0\}$, and define $\overline{\text{Record}_i} = \{1, 2, \ldots, S\} - \text{Record}_i$, and we also define $|\text{Record}_i|$ to be the number of elements in Record_i. For example, in a DTN, we assume $S = 3$, $BS = 2$, a node i has two packets $x_1 = a_1P_1 + a_2P_2$ and $x_2 = a_3P_1 + a_4P_3$ in its buffer. Thus, we have $X_i = \{x_1, x_2\}$, $|X_i| = 2$, $\text{Record}_i = \{1, 2, 3\}$, and the value of $|\text{Record}_i| = 3$, which means the encoded packets of node i including three source packets: P_1, P_2, P_3. We let Rank_i denote the rank (which equals to the value based on the definition in linear algebra) of the packets in the buffer of node i, and in both LD and PWD, to calculate Rank_i would be simply calculating $\text{Min}(BS, |\text{Record}_i|)$.

Before each transmission, a node would linearly combine all the packets in its buffer into a packet, and then transmit this packet to another node. By linear combination, the node does not need complicate forwarding strategy to achieve the even distribution of source packets. The packet being transmitted would be useful to the other node if the packet can increase its rank, or the packet can provide new source packets' information to the other node. But if the packet is useless to the other node, the result is just the other node's buffer remain the same rank after receiving current packet.

In LD approach, when two nodes meet with each other, the two nodes would use the information of their own buffers to calculate $\text{Rank}_i = \text{Min}(BS, |\text{Record}_i|)$, and transmit Rank_i packets to each other. Giving an example, let A and B denote two meeting nodes, both nodes' buffer sizes are limited to 3. Supposing all the packets in the buffers are independent, and A has three packets and B has two packets originally. When the two nodes meet, we assume B sends packets to A. Initially, B's rank is two, so it transfers two linearly combined packets of the two packets in B's buffer to A. In this example, the transmitted packets are $a_{1,5}P_1 + b_{1,5}P_2$ and $a_{1,6}P_1 + b_{1,6}P_2$. But A's buffer is already full, so A randomly do 1-to-1 combination using the incoming packets and one of the packets in A's buffer. In our example, the incoming packets are combined with the P_1 and $a_{1,2}P_1 + b_{1,2}P_2 + c_{1,2}P_3$, respectively, in A's buffer. When A is transmitting to B, A's rank is three, so it transfers three linearly combined packets to B. Then, B will combine the receiving packets with the packets in its buffer. The results would be the same as in Figure 1. LD is easy to be implemented and advantageous when buffer size is small because it does not need controlling packets before transferring.

Figure 1 Example of LD approach with $BS = 3$



In PWD approach, before starting transmitting packets, a node i would first transfer an information packet including $|X_i|$ and Record_i. We denote two meeting nodes as A and B. If B transfers packets to A, B's information packets would be useful to A in the following cases:

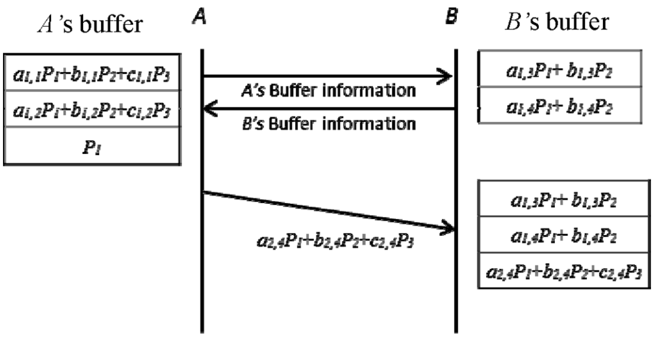
Case 1: $|\overline{\text{Record}_A} \cap \text{Record}_B| > 0$.

This is similar to it in ERP for deciding whether node B has additional information for A. So, node B would transmit $\text{Min}(|\overline{\text{Record}_A} \cap \text{Record}_B|, \text{Rank}_B)$ linearly combined packets of all the packets in B's buffer to A. If in B, $|\overline{\text{Record}_A} \cap \text{Record}_B| < \text{Rank}_B$, node B would transmit $|\overline{\text{Record}_A} \cap \text{Record}_B|$ packets to A, since node B can at most increase $|\overline{\text{Record}_A} \cap \text{Record}_B|$ ranks to A, and additional packets would not provide any new information to A. Using this heuristic, B only transmits $\text{Min}(|\overline{\text{Record}_A} \cap \text{Record}_B|, \text{Rank}_B)$ packets, other than in LD that nodes would transmit BS packets to the meeting node when buffer is full, no matter the transmitting packets can gain the ranks of the meeting node or not. PWD obtains more gain over LD when the buffer size is large, since PWD does not blindly transmit all the packets, but transmit the exact number of packets that the other node needs.

Case 2: $|X_A| < |\text{Record}_A|$ and $|X_A| < BS$.

In this case, *A* has insufficient ranks, so no matter what packets *B* have received, encoded packets from *B* can bring benefit to *A* since it can gain *A*'s rank. So in Case 2, *B* will transfer $\text{Min}(|\text{Record}_A| - |X_A|, \text{Rank}_B)$ packets to *A* to make up the insufficient rank in *A*. For example, let $BS = 4$, $|\text{Record}_A| = 4$, and $|X_A| = 2$, this means *A*'s coding matrix has four encoded source packets being combined in encoded packets, but has only two encoded packets. In this example, if $|\text{Rank}_B| >= (|\text{Record}_A| - |X_A|) = 2$, no matter what packets are encoded in *B*'s transmitting packet, transmitting two packets from *B* would increase two ranks to *A*. Other than these two cases, transmitting packets from *B* to *A* is useless for *A*. In our protocol, we would first check Case 1 then Case 2. Considering the example in Figure 2, both buffer sizes of nodes *A* and *B* are limited to three. When the two nodes *A* and *B* meet, both nodes exchange the information of their buffers.

Figure 2 Example of PWD approach with $BS = 3$



After that, assuming *B* sends packets to *A* first, *B*'s rank is two, and after testing Case 1 and Case 2, *B* finds that it cannot increase *A*'s rank by transmitting packet to *A*, so *B* does not transmit any packet to *A*. And when *A* is transmitting packets to *B*, *A* finds that it has one encoded source packet P_3 which is not in *B*'s buffer, so *A* can benefit *B* by transmitting one packet to *B* by testing Case 1, and when *A* tests on Case 2 and finds *B* has three packets in buffer and three kinds of partial packets, which means *B*'s rank is full and does not fulfil Case 2, so *A* will only transmits one linearly combined packet to *B*. In this example, PWD only transmits one data packet plus two control packets, but LD transmits five data packets.

3.2 *Multiple-packet network model*

The difference between multiple-packet network model and single-packet network model is that new packets are periodically being generated in source nodes, at any moment, older packets have been existed longer and also have more copies in the network, so they should be replaced by the newer packets in a node's buffer when its buffer is full for the effectiveness of packets. So the newer packets have higher priority than the older ones in the network.

We define same generation of packets to be a collection of packets being generated in same interval on the timeline. Packets in the same generation should have the same probability to reach the destination node at any time, so the dropping strategy should base on generation first but not the individual packet. Especially for NC-LB, if new packets can be encoded with old packets, it will be hard to decide which packet should be dropped when buffer is full, also the *decoding matrix* in the destination would be very large, so that destination would need more time to decode source packets. Here, we define the interval between the two continuous packets as generating period, and the packets that are generated roughly in the same period are in the same generation. We further define *generation*$_i$ to be the *i*th generation since the network starts working. Nodes would have many generations of packets in their buffers. Because the packet generation rate is low in DTNs, the nodes would only perform coarse-grain time synchronisation with each other.

3.2.1 *ERP for multiple-packet network model*

In transmitting stage, ERP for multiple-packet network model is the same as the ERP for single-packet network model, that one node would transmit the packets that the meeting node does not have. However, in receiving stage, when the buffer of a node is full, it would first sort the packets in its buffer according to packets' generations from old to new. If there is only one packet from the oldest generation, then the oldest generation packet would be dropped; if there are multiple packets from the oldest generation, ERP for multiple-packet network model would apply MOFO dropping strategy to the oldest generation as it in single-packet network model.

3.2.2 *NC-LB for multiple-packet network model*

For NC-LB of multiple-packet network model, in transmitting stage, we apply both LD and PWD strategies to each generation of packets in local buffer and the receiver's buffer to decide how many packets of each generation should be sent. In receiving stage, when a node with full buffer receives a new packet, rather than simply dropping an old packet as it in ERP for multiple-packet network model, it would try to linearly combine the two oldest packets which were in the same generation. In other words, in multiple-packet network model, we can apply the strategies of single-packet network model in receiving and transmitting stage to the packets from the same generation. Although the packets of the old generation would be gradually decrease and disappear from the network, NC-LB can prolong the life time of packets in the network.

Algorithm 1 illustrates the procedure in NC-LB when a node *i* meets another node. Giving an example of NC-LB for multiple-packet model, let the buffer size limit is 4, and for each packet $x_{i,j}$, *i* denotes the generation number of the packet, and *j* denotes the packet *ID*. Each packet can either be a source packet or a coded packet encoded by the source packets from the same generation. Let node A has four packets: $x_{2,a}$, $x_{3,a}$, $x_{3,b}$, $x_{5,a}$, and node B has also four packets: $x_{1,a}$, $x_{3,c}$, $x_{4,a}$, $x_{6,a}$. After using PWD strategy, B finds that every packet in B's buffer is innovative to A, so it would transmit the four packets to A. If A first receives $x_{1,a}$ from B, A would drop it since its buffer is already full, and $x_{1,a}$ is older than its oldest packet. And when A receives $x_{3,c}$, it would linearly combine $x_{3,a}$ and $x_{3,b}$ into one packet $x_{3,d}$ and stores $x_{3,c}$ in its buffer. At this time, A has $x_{2,a}$, $x_{3,c}$, $x_{3,d}$, $x_{5,a}$ in its buffer, and when A receives $x_{4,a}$, it would combine $x_{3,c}$ and $x_{3,d}$ into $x_{3,e}$ and store $x_{4,a}$ in its buffer, and A' buffer would have $x_{2,a}$, $x_{3,e}$, $x_{4,a}$, $x_{5,a}$. And if A receives $x_{6,a}$, it would drop $x_{2,a}$ because A cannot find any two packets that are in the same generation, so it will drop the oldest packet, $x_{2,a}$, and stores $x_{6,a}$ in its buffer. At last, A's buffer has $x_{3,e}$, $x_{4,a}$, $x_{5,a}$, and $x_{6,a}$. There are BS generations in buffer at most.

The packet delivery rate of ERP and NC-LB in the multiple-packet network model is closely related to the converging time in single-packet network model. Because if the destination node requires a little time to collect all the source packets, the destination node has higher probability to collect all the source packets of a generation before they are dropped in each relay node. So NC-LB would also have higher packet delivery rate than ERP in multiple-packet network model.

Algorithm 1 The overall procedure of NC-LB in multiple-packet network model

Parameters:

P_{oldest} : oldest packet in the buffer

$generation_{oldest}$: the generation number of the incoming P_{oldest}

PacketList : the list of packets in buffer, and which is sorted from old to new

$G(i)$: the generation number of the i^{th} packet in the *PacketList*

Input:

$P_{incoming} \leftarrow$ new packet from the meeting node

$generation_{incoming} \leftarrow$ the generation number of the incoming $P_{incoming}$

```
1   If buffer is not full
2       Put P_incoming into the buffer and finish the procedure
3   Else /* Buffer is full */
4       If genration_incoming is older than generation_oldest
5           Drop P_incoming
6       Else
7           Insert P_incoming to the PacketList
5           Variable j ← 0
6           Repeat j = j + 1 Until j = BS -1
7               If G(j) = G(j+1)
8                   Linearly encode the j^th packet with the j+1^th packet into a packet and put it to buffer
9                   Break
10          End Repeat Loop
11          If none of the packets were encoded together
12              Drop the packet corresponding with T_oldest
13          End If
14          If the P_incoming hasn't been encoded with any packet in step 8
15              Insert P_incoming to the buffer
16          End If
17      End else
18  End else
```

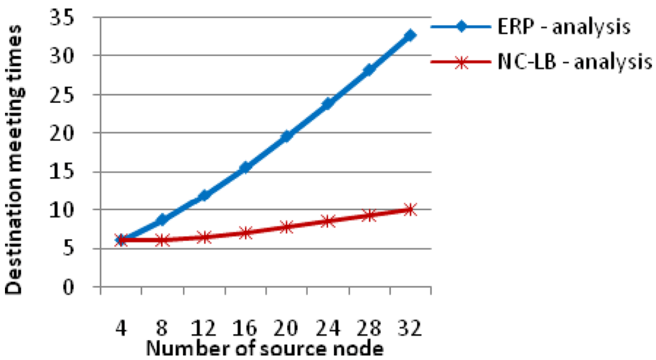
3.3 Performance analysis

In general multiple-packet network model, nodes move randomly and every node has approximately the same probability to encounter the other nodes. For a scenario with relay nodes, only source nodes contain packets in buffers at beginning, but the relay nodes have no packets. We assume that the relay nodes can meet the source nodes and obtain packets in the first few rounds. Moreover, We suppose that t rounds are necessary for every node to receive at least one packet, t must satisfy: $S{*}2^t = S + R$, so t would be $\log_2(S + R) - \log_2 S$. After t rounds meetings, the nodes need additional $\log_2 BS$ meetings to make their buffers full. So the expected converging time is at least $\log_2(S + R) - \log_2 S + \log_2 BS$ for both ERP and NC-LB.

The authors in Koetter and Médard (2003) show that the probability that a coded packet is innovative to another node is $1 - 1/q$, where q is the size of the Galois Field for generating random coding coefficients. For example, if each constant in coding vector has a size of one byte, then the field size is 8 and $q = 2^8$, thus the probability that a coded packet is innovative to another node is $1 - 2^{-8} = 0.996$, which is very close to 1. Even if the field size is 2, i.e., the transmitter is performing XORs among randomly selected packets from the pool of S source packets, the destination node will need no more than $S + 2$ coded packets on the average for decoding. On the basis of the researches, we can conclude that every packet received by relay nodes and the destination node are innovative with very high probability by using network coding. However, in ERP, nodes can double the amount of innovative packets in their buffers in a meeting with high probability only if S to BS ratio is very high. As a result, to estimate the meeting time needed to collect all the source packets, we can consider that all the packets received by destinations are innovative in NC-LB. Then the expected converging time would be $\frac{S}{BS} + \log_2(S + R) - \log_2 S + \log_2 BS$. However, it is not able to be satisfied in ERP because there are many redundant meetings without efficient packets exchange in ERP.

By the result of computing analysis in scenarios with different source nodes number, as shown in Figure 3, we can conclude that the NC-LB requires much less converging time comparing to ERP when the ratio of source nodes number to buffer size increases.

Figure 3 Converging time of different approaches under varying source numbers (see online version for colours)



4 Simulation results

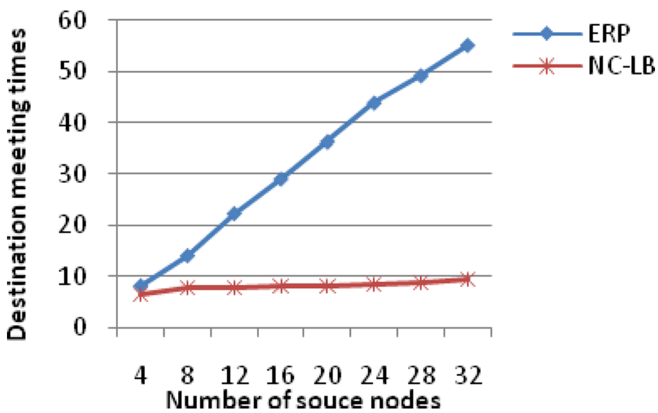
We will first show the simulation of single-generation network model, from which we can inspect the performance of NC-LB and ERP under different scenarios. Secondly, we will compare ERP and NC-LB under multiple-packet network model. We used the ONE simulator to simulate ERP and NC-LB in DTNs. We set the mobility model in simulation to be random-way point model, the simulation region is 1500 × 1500 m, and transmission range is set to 50 m. There is one destination node and 32 source nodes in the simulation. Each result data displayed following is the average over 50 simulations.

4.1 Simulation of single-packet network model

To simulate in single-packet network model, we let each source node generates only one packet at the beginning of the simulation. We let nodes move with the random-way point model (Johnson and Maltz, 1996), when nodes go into each other's transmission range, they will start to transmit. The simulation will end when the destination node collects all the source packets. The performance metric in single-packet network model in DTNs is the converging time, the time interval to collect all the source packets for destination.

Figure 4 displays the simulation results of ERP with MOFO dropping strategy, and NC-LB. It is shown when the number of source nodes increases, which means the destination needs to collect more and more source packets. In ERP, the converging time increases larger than in NC-LB. NC-LB outperforms ERP by 74.8% in average. Also, the simulation shows when $S \leq BS$, such as $S = BS = 4$, buffer is always sufficient to receive new packets, so there would be no occurrence of dropping. Thus NC-LB has no gain over ERP in this case. However, when $S > BS$, by NC-LB, we can decrease much more converging time than by ERP.

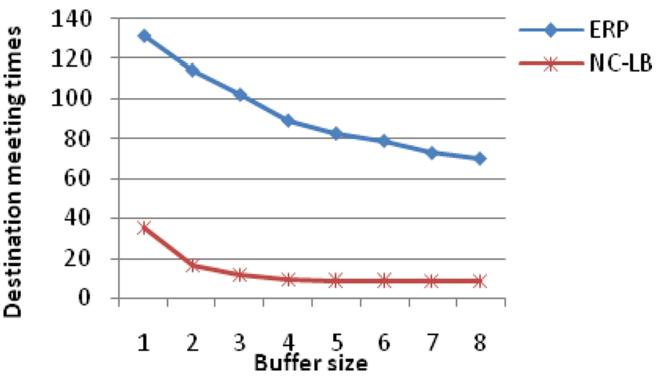
Figure 4 Converging time of ERP and NC-LB under different source nodes number (see online version for colours)



Next, we will show the influence of different buffer sizes. In Figure 5, it is shown that the converging times of NC-LB outperform its of ERP by 85.5% in average. We can conclude that the converging time under both ERP and

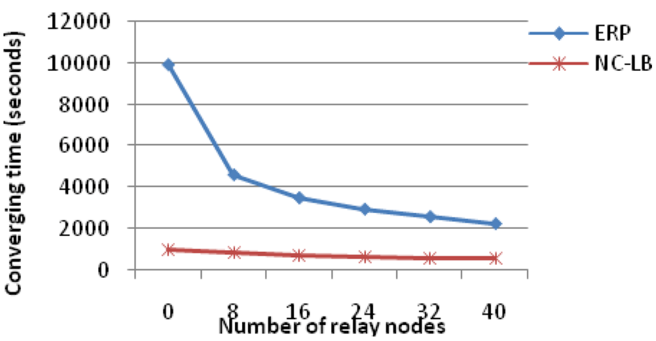
NC-LB definitely decrease with the increasing of buffer size. Still we can observe from Figure 5 that even when buffer size is 1, the destination in NC-LB needs ~35 meetings only, and when buffer size ≥ 4, the converging times stay still around 10. It means that four buffers are enough for NC-LB under this simulation environment. Moreover, the converging time would not decrease even if the buffer size continues to increase. We can also conclude from Figure 5 that with NC-LB, the buffer size requirement is much less than ERP to achieve the same converging time. Thus, NC-LB is exceptionally suitable to be the solution for DTNs with limited buffer size.

Figure 5 Converging time under different buffer size (see online version for colours)



In Figure 6, it is shown that the converging times of NC-LB outperform its of ERP by 82.6% in average under different relay nodes number. Moreover, it is shown that the converging time drastically decrease under ERP with the help of relay nodes. However, NC-LB can achieve less converging time with the help of less relay nodes than ERP.

Figure 6 Converging time of ERP and NC-LB under different relay nodes number (see online version for colours)

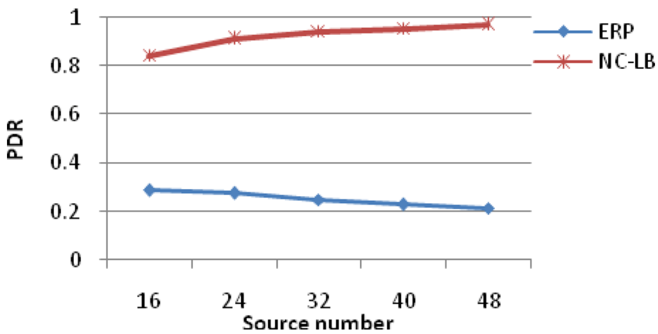


4.2 *Simulation of multiple-packet network model*

In the simulation of multiple-packet network model, packets are periodically being generated. Thus old packets should be replaced by the new generated packets. In these scenarios, the main performance metrics is packet delivery rate (PDR) instead of the time interval to collect all the source packets for destination, because destination will continue to collect multiple generations of source packets during the sensing and transmission in the network.

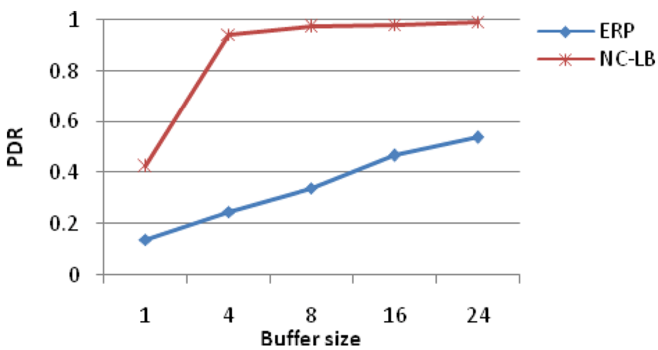
In Figure 7, it is shown that PDR of NC-LB increases, being close to 100% gradually, while PDR of ERP is less than 40%. Moreover, we can conclude that the PDR increases in NC-LB while it decreases in ERP when source node number increases. The reason is that the larger the density of source nodes in the network is, the greater the number of packets it needs to collect in each generation is. Moreover, the utility of each buffer space in ERP is not as high as in NC-LB, so with more source packets are being generated in each generation, the PDR of ERP decreases.

Figure 7 PDR of ERP and NC-LB under different source number (see online version for colours)



The influence of the buffer size is shown in Figure 8. PDR of ERP increases steadily with the increasing of buffer size. However, PDR of NC-LB increases more quickly than it in PDR and keeps in a higher level of being close to 100%, while PDR of ERP is <60%. Moreover, as the result shown in Figure 5, NC-LB with 4 buffer space is enough to achieve high PDR in this environment.

Figure 8 PDR of ERP and NC-LB under different buffer size, $S = 32$ (see online version for colours)



5 Conclusions

In DTNs, the long transmission delay of packets will decrease the efficiency and availability of applications. In this paper, we proposed a network coding with limited buffer routing protocol, NC-LB, which includes the linearly random network coding technique to enhance the performance in communications. Nodes encode packets to save space while keeping the partial information of each packet, and nodes also encode all the packets in their buffers before transmitting, which makes the encoded packets innovative to the other nodes with high probability.

In simulation, it is shown that the destination in NC-LB needs less time in average to collect all the source packets than ERP in single-packet network model by different source nodes number, buffer size and relay nodes number. Moreover, the advantage of NC-LB becomes apparent especially when the ratio of source to buffer size is higher, and NC-LB can save more than 70% converging delay than ERP. We also implemented the simulation in multiple-packet network model, it is shown that due to the less requirement of packet collecting time, NC-LB has higher data delivery rate over ERP by different source nodes number and buffer size.

References

Abdulla, M. and Simon, R. (2008) 'Controlled epidemic routing for multicasting in delay tolerant networks', *Proceedings of IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, Baltimore, MD, USA, pp.1–10.

Ahlswede, R., Cai, N., Li, S.Y.R. and Yeung, R.W. (2000) 'Network information flow', *IEEE Transactions on Information Theory*, Vol. 46, pp.1204–1216.

Chou, P.A., Wu, Y. and Jain, K. (2003) 'Practical network coding', *Proceedings of the 41st Annual Allerton Conference on Communication*, Control and Computing, Monticello, USA.

Fall, K. (2003) 'A delay-tolerant network architecture for challenged internets', *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Karlsruhe, Germany, pp.1–14.

Fathima, G. and Wahidabanu, R. (2011) 'Integrating buffer management with epidemic routing in delay tolerant networks', *Journal of Computer Science*, Vol. 7, pp.1038–1045.

Ferrari Aggradi, G., Esposito, F. and Matta, I. (2008) 'Supporting predicate routing in DTN over MANET', *Proceedings of the Third ACM Workshop on Challenged networks*, San Francisco, California, USA, pp.125–128.

Hossmann, T., Spyropoulos, T. and Legendre, F. (2010) 'Know thy neighbor: Towards optimal mapping of contacts to social graphs for DTN routing', *Proceedings of the 29th Conference on Information Communications*, San Diego, USA, pp.1–9.

Jin, T., Wang, Y.F. and Random, F.Z. (2011) 'Linear network coding with cooperating in wireless sensor network', *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, China, pp.1–4.

Johnson, D.B. and Maltz, D.A. (1996) 'Dynamic source routing in ad hoc wireless networks', in Imelinsky, T. and Korth, H. (Eds.): *Mobile Computing*, Kluwer Academic Publishers, pp.153–181.

Koetter, R. and Médard, M. (2003) 'An algebraic approach to network coding', *IEEE/ACM Transactions on Networking*, Vol. 11, pp.782–795.

Li, Q., Zhu, S. and Cao, G. (2010) 'Routing in socially selfish delay tolerant networks', *Proceedings of the 29th Conference on Information Communications*, San Diego, USA, pp.1–9.

Li, Z., Zeng, D.Z., Guo, S., Lu, S.L., Chen, D.X. *et al.* (2012) 'On the throughput of feedback less segmented network coding in delay tolerant networks', *IEEE Wireless Communications Letters*, Vol. 1, pp.93–96.

Lidl, R., Niederreiter, H. and Cohn, P.M. (1997) *Finite Fields*, Vol. 20, Cambridge University Press, Cambridge, UK.

Lin, Y., Li, B. and Liang, B. (2008) 'Stochastic analysis of network coding in epidemic routing', *IEEE Journal on Selected Areas in Communications*, Vol. 26, pp.794–808.

Lindgren, A. and Phanse, K.S. (2006) 'Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks', *Proceedings of First International Conference on Communication System Software and Middleware*, Delhi, India, pp.1–10.

Lu, X. and Hui, P. (2010) 'An energy-efficient n-epidemic routing protocol for delay tolerant networks', *Proceedings of the 2010 IEEE Fifth International Conference on Networking, Architecture, and Storage*, Macau, China, pp.341–347.

Motwani, R. and Raghavan, P. (2010) *Randomized Algorithms*, Chapman & Hall/CRC, New York, USA.

Qin, S. and Feng, G. (2013) 'Performance modeling of network coding based epidemic routing in DTNs', *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2013)*, Shanghai, China, pp.2057–2062.

Vahdat, A. and Becker, D. (2000) 'Epidemic routing for partially connected ad hoc networks', *Technical Report CS-200006*, Duke University.

Yoon, S.K. and Haas, Z.J. (2010) 'Application of linear network coding in delay tolerant networks', *Proceedings of the 2010 Second International Conference on Ubiquitous and Future Networks*, Jeju Island, Korea, pp.338–343.

Yunfeng, L., Baochun, L. and Ben, L. (2008) 'Efficient network coded data transmissions in disruption tolerant networks', *Proceedings of the 27th IEEE Conference on Computer Communications*, Phoenix Arizona, USA, pp.1508–1516.

Zeng, D., Guo, S. and Hu, J. (2013) 'Reliable bulk-data dissemination in delay tolerant networks', *IEEE Transactions on Parallel and Distributed Systems*, 28 August, IEEE Computer Society Digital Library.