RAPPORT DE RECHERCHE

L R I

NETWORK CODING BASED FLOODING USING FOUNTAIN CODES

KADI N / AL AGHA K

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

08/2008

Rapport de Recherche N° 1500

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

Network Coding based flooding using Fountain Codes

Nour KADI and Khaldoun Al AGHA
LRI, Laboratoire de Recherche en Informatique, Paris XI University
Email: {nour|alagha}@lri.fr

***Abstract*—Multipoint relays "MPR" have been introduced in the proactive protocol OLSR in order to optimize the flooding overhead of control traffic. In this paper we propose a protocol NFCF to optimize the Network Coding based flooding using Fountain Codes such as LT code. our protocol combine the MPR technique with the notion of network coding where MPR nodes do not simply forward packets they overhear but may send out information that is coded over the contents of several packets they received. Our interest is to reduce the number of transmissions. We show by simulation that flooding can be efficiently performed by exploiting network coding with multipoint relays, resulting in significant reductions in the number of transmissions in the network. We can get benefits from this approach also in other practical considerations such as restricted complexity and memory capabilities.**

I. Introduction

A Mobile Ad-Hoc NETwork (MANET) is a collection of mobile nodes that communicate using a wireless medium. It is considered as the ideal communications technology for scenarios where the network infrastructure is missing. A lot of challenges found at this type of network such as limited bandwidth, limited power supply and the channel condition which can vary greatly. Thus we need robust routing protocols to enable efficient communication in such network.

Flooding is a traffic configuration of the network at which all the nodes transmit information to all other nodes forming the network. This configuration plays an important role in most of the routing protocol in MANET such as AODV [1] and OLSR [2]. However flooding leads to a large amount of redundant messages that consume bandwidth and power in addition to the packet loss because of the collision. The high overhead of using naive flooding was highlighted in [3] and this increases the need of an efficient mechanism for flooding.

OLSR (Optimized Link State Routing) protocol [2] is a well-known proactive routing protocol for MANETs. It uses an efficient mechanism, called multipoint relays (MPR), for flooding control traffic which reduces the number of transmissions required. Each node selects its own MPRs set which is a set of its one-hop neighbors that cover the two-hop neighbors. In OLSR, only those nodes selected as MPR are responsible for forwarding control traffic into the entire network.

Separately, Forward Error Correction (FEC) schemes such as fountain codes [4], [5] are provided as a solution for reliable communication in lossy networks. They obviate the need of retransmission request and thus decrease the bandwidth consumption as well decrease delay. The most important characteristic of these codes is their low complexity of encoding and decoding functions. These schemes are considered as end-to-end approach where the source node generates a limitless of encoded packets, the destination when it receives enough packets, it decodes them to get the source packets, but the intermediate nodes are only allowed to replicate and forward packets and don't participate in coding. However this approach decreases the network throughput and in order overcome this drawback, the authors of [6] [7] propose an alternative approach where network coding is employed. In the context of network coding at [6] the intermediate nodes send out random linear combinations of all previously received packets. [7] investigates FEC codes at the intermediate nodes in order to reduce the complexity processing.

Network coding is a new technique that appeared for first time in the work of Ahlswede et al. [8] which showed the utility of the network coding for multicast. The notion of network coding means that the nodes can combine and mix the packets rather than merely forward them. Authors in [9] showed that codes with a simple, linear structure were sufficient to achieve the capacity of multicast connections in lossless, wired networks. Network coding has been shown to be useful in the context of wireless communications for several configurations [10], [11], [12], [13], [14]. However, despite the importance of flooding, little research efforts were made to employ network coding for flooding. We presented in our previous work [15] an approach that integrates the MPR technique with notion of network coding in order to optimize flooding in wireless ad hoc network. We proposed an algorithm that employs an opportunistic coding at each MPR node based on simple XOR operations.

In this paper we optimize our coding algorithm using fountain code. This paper aims to provide a practical network design and provide an operational protocol for flooding. The proposed solution include a simple encoding algorithm based on the principle of LT code to perform the opportunistic coding while taking into account the delay, complexity, reliability and memory requirement.

Our design uses the following principle:

- It is based on deterministic flooding as it employs MPR nodes which are the only nodes in the network responsible for rebroadcasting the packets. This technique reduces the redundant retransmissions.
- It employs opportunistic network coding in order to overload each transmission with additional information

as possible without delaying the packets.

- It implements the coding by using the principle LT code which is rateless erasure code and is characterized by the simplicity of its encoding and decoding functions.

Our main goal is to reduce the cost of flooding. We refer to the cost as the total number of transmissions required. We show by simulation how can our design achieves this goal and also we can get benefit in term of complexity, delay and memory requirement.

II. Limitation of previous works:

Works in [16], [17], [14] seeks to find efficient broadcast methods in an ad hoc environment using network coding.They present an approach where each intermediate node performs a random linear combination of all previously received packets. If we have K packets for broadcast, so encoding process at each node need $O(K)$ operations and the decoding process, which depends on Gaussian Elimination, needs $O(K^3)$ operations. Also all of these works are probabilistic approaches as each node has a parameter called rate or retransmission factor which determines the number of coded packets that a node should broadcast when it receives a packet. Choosing to increase this rate ensures the delivery of all the packets, however in the other hand it increases the redundant transmissions which is inefficient. Finding the optimal rate of the coding nodes is a hard problem. Generations management is another important design decision in these works. Size and composition of generations may have significant impact on the performance of network coding. Distributed generation management is also a hard problem and unsuitable solution for this problem increases the decoding complexity and limits the coding gain.

CODEB presented in [18] is the only deterministic approach found which uses network coding for optimizing the broadcast in ad hoc wireless network. It use the partial dominant pruning (PDP) technique which is similar to MPR technique in selecting certain relaying nodes to rebroadcast the packets in the network. Both technique use a greedy set cover GSC algorithm for selecting the set of forwarder however the MPRs selecting algorithm chooses as forwarders those candidates that have exclusive coverage of some two-hop neighbor, and only then apply GSC over the remaining nodes [19]. This additional step permits to decrease the number of relying nodes in the network so the MPR outperform the PDP technique. Also in PDP technique the forwarder list is determined at a node for each packet to be broadcast depending on the source of the packets and so the IDs of the forwarder should be listed in the header of the packet which causes some overhead. However in MPR technique, the forwarder list (the MPRs) of a node is unique for all the packets that pass the node and there is no need to recalculate this set for each packet. For the encoding process they present two algorithms one of them is based on Reed-Solomon code. This code has many drawbacks such as: The node u should have knowledge about the packets that its neighbors lost in order to determine, k, the number of coded packets that should be sent over the n native packets. And then these neighbors should receive certain number of these encoded packets to start decoding and getting the whole batch of n native packets. If any of these coded packets is lost during the transmission, a retransmission is required and this retransmission could be redundant for other nodes. If u receive new packets before finishing the transmission of the k encoded packet, so the new packet would be buffered until all neighbors receive the precedent batch of the n native packets or a recalculation of the whole encoded packets is required. Also in case of mobility this scenario is not efficient. In addition to the limitations of building a digital fountain with Reed-solomon codes as explained in [4] [20] [21]. As the number of packets increases, the cost of Reed-Solomon coding induce too much overhead.

[22] presents FBcast which combines erasure code with probabilistic broadcast technique for reliable broadcast. However at this approach they use end-to-end coding which means that the encoding process is done only at the source node and then the intermediate nodes only forward the received packet. This can reduce the network capacity as referred in [6]. The authors of [7] show that min-cut capacity can be achieved if we allow intermediate node to process the incoming packets and perform coding. They present several coding schemes however this work is mainly theoretical without providing any practical design. So our aim is to bridge theory with the practice and avoid the limitations of the previous works.

III. Requirement for efficient flooding and the proposed solution:

Let's take an example of a proactive routing protocol such as OLSR, where flooding neighbourhood information is required in order to build the routing tables at each node. We assume that each node has a packet, which contain the topology control message, should be delivered to the entire network. The following challenges should be taken into account for an efficient protocol for flooding these packets:

- Cost: the number of transmissions required should be minimal.
- Scalability: Network traffic overhead is manageable even for very large number of flooding packets.
- Delay: the packets should be delivered with minimum delay.
- Memory requirement: the amount of memory required to keep the received packets for rebroadcasting should be minimal.
- Mobility-independent: Nodes could move freely in the network and their neighbors continue sending their set of packets without additional overhead for sending more redundant packets.
- Time-efficient: the amount of processing required at each node for both sending side and receiving side is minimal.

We provide a practical design for a deterministic approach of flooding in wireless ad hoc network. Our approach is based on MPR technique. MPR technique reduces the duplicate retransmissions and thus reduces the flooding cost. In the other hand the flooding cost could be reduced more by employing network coding which overload each transmission with additional information. In order to reduce the complexity of coding and decoding at each node, we employ the principle of fountain code, such as LT code, to perform network coding. Also using LT code can reduce the delay as the decoding process is performed step by step such that each received encoded packet could release several native packets and so we use these native packets for re-coding and rebroadcasting. So the node doesn't have to wait until receiving the whole batch of the encoded packets. For that we can use the opportunistic coding which means that whenever a node has an opportunity to transmit a packet (MAC layer access), it chooses packets to encode among the already received and decoded packets without waiting any additional packets to be received or decoded. Moreover, as each encoding packet can be generated independently of other encoding packets and so decoded separately so the node can receive the encoded packets from several neighbors because the native packets could be retrieved from any set of coding packets. Thus the node could move freely through the network and continue receiving the packets from any neighbors without loosing any already received packet. Finally, in the case of lossy network, no retransmission is required as using the fountain code itself provide the reliability.

IV. Background and previous Approach

In this section we first formally introduce the problem formulation. Then as this paper departs from our previous work [15] so for completeness we briefly review our previous approach here.

A. Problem Formulation

As mentioned earlier that in our approach instead that MPR node sends one single native packet, it can encode several packets together and so it delivers more than one packet in single transmission. Now we give a description to the algorithm that finds the coding candidates.

We want to find the maximum number of packets that could be decoded from maximum number of neighbors. If P is an encoded packet where $P = \oplus_{i=0}^{K} p_i$, we call CCL the coding candidates list.

$$CCL = \{p_0, p_1, \cdots, p_K\}$$

Let $F(P)$ be a function that gives the number of neighbors that can decode P and $g(P)$ is a function that represents the delay constraint which is, in our simulation, the average waiting time of the coding candidates in the queue.

$$g(x) = Avg_{p_i \in CCL}\{del(p_i)\}$$

Now consider a node u which has N packets in its output queue and has M neighbors. We refer to the output queue by Q.

$$Q = \{P_0, P_1, \cdots, P_N\}$$

So our problem is to find $P \in C$ such that maximize $F(P)$ while minimize $g(P)$, where C is a set of all possible combinations of the packets in Q.

$$C = \{\oplus_{i=0}^{K} PP_i \quad ; \quad PP_i \in Q, K \leq N\}$$

In the other hand if a neighbor of u receives P, it can decode it and get a new native packet if and only if it has at least $K-1$ packets from the list CCL. From this decoding rule we can conclude that the maximum number of packets that could be decoded from all the neighbors is M. So $K \leq M$. As mentioned earlier, we want at the same time to optimize the delay of the packet, such that at each transmission we send at least one packet which has the longest delay in the queue. For this reason we always pick the packet at the head of the queue to add it to the coding candidate list. So $PP_0 = P_0$. In order to find the optimal solution of such problem, we have to test all the possible combinations of the packets in the queue and find for each combination the number of neighbors that can decode. Then pick the combination that could be decoded at the maximum number of neighbors. However to do that the number of possible combination that we have to test is

$$C_N^1 + C_N^2 + C_N^3 + \cdots + C_N^{M-1} = N + \frac{N(N-1)}{2!} + \cdots + \frac{N(N-1)\cdots(N-M-2)}{(M-1)!} \approx O(N^{M-1})$$

which is too expensive.

B. Previous Result

In [15] we introduced a new protocol for ad hoc wireless network which integrates the MPR technique with the notion of network coding. We proposed a simple distributed algorithm for opportunistically choosing packets to encode and broadcast. The encoding function is based on simple XOR operation. We review our protocol as follow:

1) Protocol overview: As in OLSR, nodes periodically broadcast HELLO messages to their neighbors. These messages contain the list of one hop neighbors of the node with their link status. They permit the node to calculate its one-hop and two-hop neighbor set. Thus the node can select its MPR set by selecting the minimal number of one-hop neighbors which covers all its two-hop neighbors. Finding the smallest multipoint relay set has been shown to be NP hard [23]. So we propose to use the heuristic mentioned in [24].

Separately, each node maintains the flowing two queues to keep the packets:

- *Input queue:* The node keeps in this queue the packets that have been received in addition to the neighbor reception information that gives the probability of each neighbor having that packet.
- *Output queue:* It is FIFO queue where the node keeps the packets to be broadcasted until they will be delivered to all the neighbors.

Inserting a packet in the output queue follows the rule *"A node inserts a new native packet in the output queue to rebroadcast only if it has received its first copy from a node for which it is a multipoint relay"*. When a node has an opportunity to send, signalled by MAC, it checks its output queue to find the best set of packets to encode together, then broadcast the encoded packet. The encoded packet is obtained by simply XORing those chosen packets together and reporting their IDs in the header. When a node overhears a packet, it decodes the encoded packet, if it is possible, and stores the new packet in the input queue for a limited period T. In addition the node sends an acknowledgement to its neighbors to confirm its reception of the packet and its ability to decode it. Because of the broadcast nature of the wireless medium, this acknowledgement will be received by all the neighbors and used to update the neighbor reception information in the input queue. Also these acknowledgements are important to inform the node that certain packets has been received by all its neighbors and so it can delete it from its output queue.

2) Distributed Packet Coding Algorithm: In order to find the candidates of the encoding packet we proposed a heuristic to find the coding list that contains candidates from the output queue which could be decoded at maximum number of neighbors. Our heuristic needs $O(N \times M)$ operations where N is the number of packets in the queue and M is the number of neighbors. However the key point to choose good coding candidates is to know the packets which exist at each neighbor. As explained earlier, we propose to associate each packet in the input queue with neighbor reception information. These information are updated according to the reception of the acknowledgements from the neighbors. Moreover, this information could be updated according to the two-hop neighborhood information, obtained by HELLO messages, and the previous hop of the received packet.

3) Receiving and Decoding the packet: When the node u receives a native packet, it inserts it directly in the input queue and if this packet is received from one of the MPR selector of u then it inserts it into the output queue. However in case it receives an encoded packet, u should extract the new native packet before inserting this new packet in the queues. So u will decode the packet, if it is possible, by simply XORing it with the known native packets stored in the input queue. Finally, it schedules an acknowledgment to inform its neighbors about the new native packet that it has received.

C. Simulation Results

Our simulation results demonstrated that our previous approach result significantly reduce the number of transmissions in the network. We can get benefits from this approach also in other practical considerations such as restricted complexity and memory capabilities. However the main drawback that could be found at this approach is that the coding algorithm depends on learning the neighbor state. So any loss or delay of the acknowledgements from the neighbors cause the neighbor reception information to be inexact and thus the coding choice will not be efficient. So in order to avoid this problem and to optimize our distributed coding algorithm we propose to combine one of the fountain code which is LT code with the network coding as explained the following section.

V. NCFC Overview

We introduce NCFC, a new broadcast protocol to optimize the **N**etwork **C**oding based flooding using **F**ountain **C**odes. This approach combines the MPR-based flooding with opportunistic network coding.

In this approach we try to optimize MPR-Based flooding using LT Code which is one of the digital fountain codes. LT code is a rateless erasure code as the source generates limitless coding symbols until the destination can decode these symbols and recover the source data when it receives sufficient number of encoding symbols. So in this way the source don't need to fix its rate during the transmission, instead it sends as much coding symbols as needed. And also the source doesn't need to know the neighbors state (the packets that each neighbor has) because the encoding symbols are generated randomly. Integrating the two techniques together which are MPR technique and network coding technique, represented by LT code, can effectively reduce the cost of flooding because MPR reduce the redundant transmission and the network coding overload each transmission with additional packets as possible.

A. How does our approach work?

Each node has three buffers:

- I *ReceivedPackets buffer:* keeps ”for certain time” all the native packets, which are not encoded, that the node has received or recovered from the decoding process.
- II *PacketToBroadcast buffer:* keeps the packets that the node had received and waits for broadcast. When the packet is received by all the neighbors, it should be deleted from this buffer.
- III *EncodedPackets buffer:* keeps the received encoding packets that couldn't be decoded immediately.

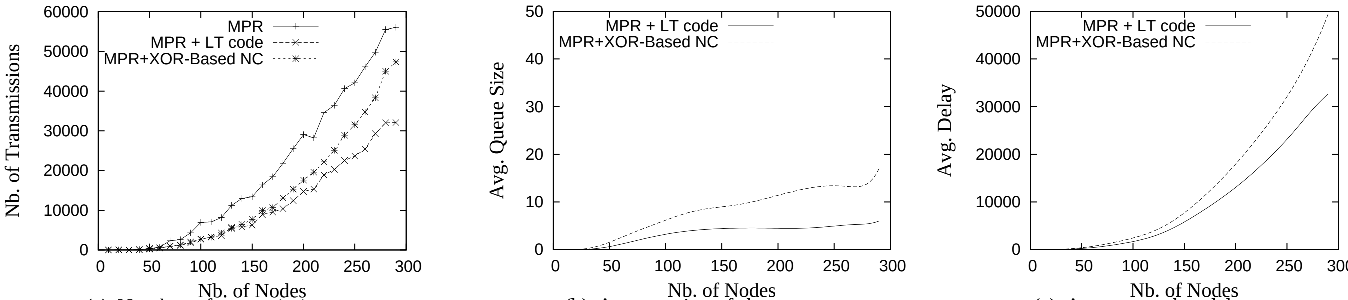
As we use MPR-based flooding, so the packets in buffer II are only those which are received from the MPR selector of the node.

B. Sender Side:

Each time a node has a sending opportunity signalled by MAC, and if it has any packets in buffer II, it uses LT code to find a set of packets from the buffer II to encode together and broadcast it to its neighbours.

The encoding technique in LT code is as follow:

1) Choose randomly a degree d of the encoded packet which represents the number of packets to encode. This degree is chosen according to ideal soliton distribution.
2) Choose d packets randomly from the buffer II to use them as coding candidates, and XOR these coding candidates to get an encoded packet.
3) Add the ids of coding candidates into the header and then broadcast the encoded packet.



(a) Number of transmissions (b) Average size of the queue (c) Average packet delay

Fig. 1. performance comparison of MPR-based coding and OLSR and only-network-coding-based flooding.

C. Receiver Side

The node needs to retrieve the native packets from the received packet by using the decoding process as follow:

1) If the received packet R is of degree $d > 1$, the node tries to reduce this degree by processing the packet R against all native packets in buffer I. That means to XOR the packet R with all the native packets stored in buffer I which are coding candidates of R. Each coding candidate found in buffer I reduce the degree of R by 1. If the reduced degree of R is still greater than 1, it will be stored in buffer III.
   For example if a node u has 2 packets P_1, P_2 in buffer I. then it receives an encoded packet $R = P_2 \oplus P_3 \oplus P4$ of degree 2. By XORing P_2 with R we get $R' = R \oplus P_2 = P_2 \oplus P_3 \oplus P_4 \oplus P_2 = P_3 \oplus P_4$ So R' has degree 2
2) If the received packet R has a degree 1 or its degree has been reduced to 1 in the previous step, it is inserted into buffer I as it is considered as native packet. And then this native packet is used to reduce the degree of the other encoded packets by matching it against all the encoded packets residing in buffer III.
3) When the previous step successes to reduce a degree of any packet in buffer III to degree 1, then this packet is inserted into buffer I and also it is processed against the packets remaining in buffer III.
4) The receiver schedules an acknowledgment to inform its neighbors about the new native packets which are inserted into buffer II after this process. This acknowledgement could be piggybacked with another data packet.

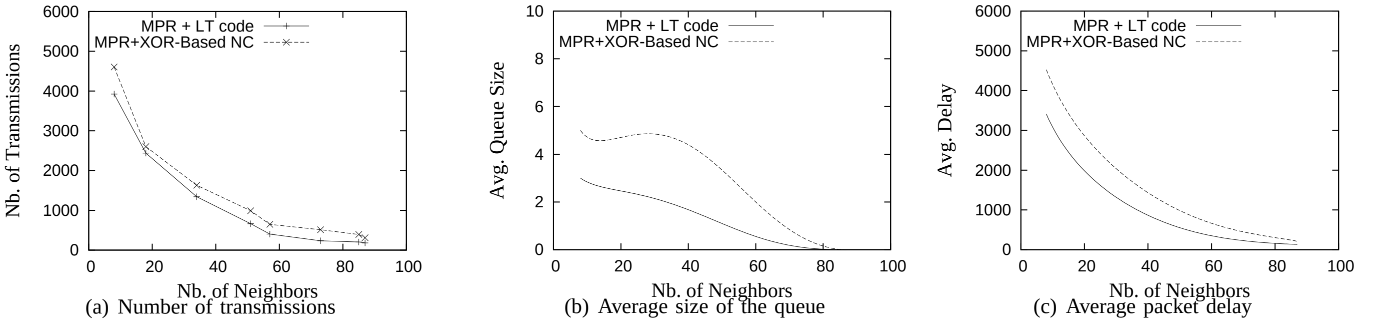
Any native packet received or recovered during the decoding process is inserted into buffer II if and only if it has received from a MPR selector of a node. And this packet remains in this buffer and has a chance to be broadcasted until it is received by all the neighbors of the node. This is signalled by the acknowledgements sent by the neighbors.

VI. Simulation Results

To evaluate the efficiency of our NCFC algorithm, we compare its performance with our previous approach that combine the MPR technique with simple XOR algorithm to perform network coding at each MPR node. In [15] we show how our previous approach outperforms the other protocol that use either MPR technique or network coding separately. In this section we show that the optimization performed over the algorithm of coding by using LT code can improve the performance and reduce the flooding cost. Our performance metrics are the number of transmissions needed for flooding, memory requirement and the packets delay. To evaluate the memory requirement we calculate the average size of the buffers. For the previous approach we calculate the size of the output queue where a node keeps the packets that should be rebroadcasted until MAC indicates a send opportunity. For NCFC we calculate the sum of the size of the two buffers: the PacketToBroadcast buffer and the EncodedPackets buffer. And for the evaluation of the packets delay we use the average time needed for the packets to be delivered to the entire network. We implement the protocols in a network simulator. Nodes are placed randomly on the simulation area. Transmissions are received by all nodes within transmission range. A packet transmission takes exactly one time unit. A node can either send or receive only one packet at a time unit. For network traffic, we assume that each node has one packet to broadcast to all the nodes in the network. Then, the simulation continues to run without inserting further packets until all the packets delivered to the entire network. At each time unit we randomly pick a node and schedule its transmission. So we assume that only one node can send at a given time unit and in this way we avoid packet collisions. However a network provided with more accurate physical layer and MAC layer for perfect collision avoidance can give better result. This is because a network coding and specifically LT code increases the robustness against packet loss as a packet may be sent many times before it can be deleted from the queue. The acknowledgment that we propose plays an important role for the robustness. We first analyze simulation for different number of nodes spread in the network area where they have nearly the same density and then we see the impact of different densities on protocol performance.

A. Different Number of Nodes:

First we compare the performance for different number of nodes. We suppose that the average number of neighbors is nearly 20. The nodes are placed randomly in the network area which is chosen according to the number of nodes.



(a) Number of transmissions (b) Average size of the queue (c) Average packet delay

Fig. 2. Impact of Node Density.

The number of nodes changes from 30 to 300. As shown in figure 1(a) that NCFC reduces significantly the number of transmissions. We see that the difference in performance is most pronounced for large number of nodes and this is because when the number of nodes increases, the number of packets increases accordingly and so each MPR node of OLSR will have more packets to transmit individually. However, in the case where network coding is employed, increasing the number of packets in the queue increases the coding opportunities which overloads each transmission with additional packets and thus reduce the increasing of the number of transmissions. As shown in Figure 1(b)&(c) NCFC reduces the memory requirement and the packet delay as using network coding at MPR nodes makes the draining rate of the packets from the queues higher and this reduces the size of these queues in our approach.

B. Impact of Node Density:

The modified parameter here is the average number of neighbors, accomplished by varying the radio range of the nodes. We suppose that the network contains always 200 nodes, randomly distributed on a surface $1500m \times 1500m$. The average number of neighbors are 20, 35, 51, 75, 87. The corresponding simulation results are shown in Fig. 2 which shows that NCFC outperform the other protocol in term of cost, delay and memory requirement. As we use the MPR technique it is normal to see that the number of transmissions is reduced is dense network. And using the notion of network coding can reduce the the memory requirement and the delay of the packets as it increases the draining rate of the buffer when we send as much packets as possible at each transmission. we see from the simulation result that using the principle of LT in the encoding process gives better choice for choosing the coding candidates than using our previous XOR-based coding algorithm. As the candidates are chosen randomly and we don't restrict the encoded packet to be decoded immediately instead it could be buffered until receiving other packets. These received packets could be from any neighbor and thus accelerate the decoding process. Moreover, the interested point is that when we calculated the memory requirement for LT code approache we take into account the buffer needed to store these encoded packets that couldn't be decoded immediately and the simulation result show that even with this additional buffer the memory requirement when using LT code is still lower than that when using the XOR-based approach.

VII. Conclusions

In this paper we have presented the initial steps for practical design that combine MPR technique with Network coding to optimize flooding in wireless ad hoc network. We have develop a simple distributed algorithm that use the principle of LT code to perform network coding. LT code is one of the most popular erasure code that is characterized by its simplicity in encoding and decoding process. Our simulation result has demonstrated the performance improvement achieved using our approach.

This work is a first step towards a full communication architecture. In the future, we intend to investigate a number of issues such as NCFC performance in term of packet delivery ratio and to explore more on the reliability issue.

References

[1] C. E. Perkins and E. M. Belding-Royer, "Ad-hoc on-demand distance vector routing," in *WMCSA*. IEEE Computer Society, 1999, pp. 90–100.

[2] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," IETF, RFC 3626, October 2003.

[3] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2-3, pp. 153–167, 2002.

[4] Luby, "LT codes," in *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[5] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[6] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *CoRR*, vol. abs/cs/0510070, 2005, informal publication.

[7] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," *CoRR*, vol. abs/cs/0508124, 2005, informal publication.

[8] Ahlswede, Cai, Li, and Yeung, "Network information flow," *IEEETIT: IEEE Transactions on Information Theory*, vol. 46, 2000.

[9] Li, Yeung, and Cai, "Linear network coding," *IEEETIT: IEEE Transactions on Information Theory*, vol. 49, 2003.

[10] Y. Wu, P. A. Chou, and S. Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," Microsoft Research (MSR), Tech. Rep. MSR-TR-2004-78, August 2004.

[11] Y. W. P. A. Chou and K. Jain, "Practical network coding," in *Allerton Conference on Communication, Control, and Computing, Monticello, IL*, 2003.

[12] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *SIGCOMM*. ACM, 2006, pp. 243–254.

[13] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *SIGCOMM*. ACM, 2007, pp. 169–180.

[14] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. on Networking*, vol. 16, no. 2, pp. 450–463, 2008.

[15] N. KADI and K. AL AGHA, "Optimized mpr-based flooding in wireless ad hoc network using network coding," LRI, Tech. Rep. 1499, July 2008.

[16] J. Widmer and J.-Y. L. Boudec, "Network coding for efficient communication in extreme networks," *In Workshop on delay tolerant networking and related networks (WDTN-05)*, vol. 5, August 2005.

[17] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "A network coding approach to energy efficient broadcasting: From theory to practice," in *INFOCOM*. IEEE, 2006.

[18] E. L. Li, R. Ramjee, M. M. Buddhikot, and S. C. Miller, "Network coding-based broadcast in mobile ad-hoc networks," in *INFOCOM*. IEEE, 2007, pp. 1739–1747.

[19] M. A. Spohn and J. J. Garcia-Luna-Aceves, "Improving route discovery in on-demand routing protocols using two-hop connected dominating sets," *Ad Hoc Networks*, vol. 4, no. 4, pp. 509–531, 2006.

[20] D. Mackay, "Fountain codes," in . *IEEE proc.-commun*, 2005.

[21] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "Digital fountain approach to reliable distribution of bulk data," International Computer Science Institute, Berkeley, CA, Tech. Rep. TR-98-005, Feb. 1998.

[22] R. Kumar, A. Paul, and U. Ramachandran, "Fountain broadcast for wireless networks," in *IEEE 2nd International Workshop on Network Sensing Systems (INSS). San Diego, CA*, 2005.

[23] L. Viennot, "Complexity results on election of multipoint relays in wireless networks," INRIA, Tech. Rep. RR-3584, 2007.

[24] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," INRIA, Tech. Rep. Research Report RR-3898, Feb. 2000.