RAPPORT DE RECHERCHE

L R I

OPTIMIZED MPR-BASED FLOODING IN WIRELESS AD-HOC NETWORK USING NETWORK CODING

KADI N / AL AGHA K

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

07/2008

Rapport de Recherche N° 1499

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

Optimized MPR-Based Flooding in Wireless Ad Hoc Network using Network Coding

Nour KADI and Khaldoun Al AGHA

LRI Technical Report

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 an algorithm to optimize the MPR-based flooding by using 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.

1 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 and cause collisions and thus packet loss. [3] shows the problems associated with using pure flooding for broadcasting 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, network coding is a new technique that appeared for first time in the work of Ahlswede et al. [4] 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 [5] 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 [6], [7], [8], [9], [10], [11]. However, despite the importance of flooding, little research efforts were made to employ network coding for flooding. Works in [12], [13], [10] seeks to find efficient broadcast methods in an ad hoc environment using network coding. 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. Authors in [11] use a deterministic approach to optimize flooding using the technique of partial dominant pruning with network coding. However in this technique set of forwarder nodes should be calculated separately for each received message.

In this paper we show how network coding provides significant gains when applied to OLSR protocol where MPRs are employed to reduce the redundant transmissions. We apply the network coding only to those nodes which are selected as MPRs. We provide distributed algorithm to perform an opportunistic coding at MPR nodes. The memory available at a node constrains the number of packets a node can keep before rebroadcast them. We discuss how efficient our approach can achieve even with little available memory as integrating MPR with network coding can significantly decrease the number of packets in the buffer as well it reduces the delay of the packets in the buffer. We compare the performance of alternative protocols which use either MPR technique or network coding technique to our approach which integrates the two techniques together and find that our approach significantly outperforms the alternative protocols in term of performance, complexity, scalability and memory occupation.

The remainder of this paper is organized as follows. Section 2 motivates the MPR-based coding by an example. Our main contribution is MPR-based coding protocol and is presented in Section 3. it consists of a distributed algorithm to choose the packets to encode and another to receive and decode the packets. Simulation results are presented in Section 4.

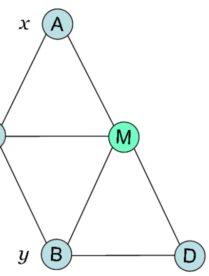


Figure 1: Motivation example

2 Motivating Example

Let's start with a simple example which illustrates the gains that can be achieved by integrating network coding with MPR. Consider the network in Figure 1, where nodes A, B have the packets x, y respectively to broadcast to all nodes in the network. Using pure flooding, we need 10 transmissions to deliver the two packets to the entire network. Using OLSR, node M is considered the multipoint relay of both A, B. So M and C receive x, y from A, B respectively and then only M will rebroadcast the two packets separately. Thus we need 4 transmissions. Using network coding, both C and M receive x, y and both perform the network coding to mix the two received packets together and send the encoding packet $z = x \oplus y$. so the nodes A, B and D should decode the received packet in order to recover x and y by simply doing either $z \oplus x$ or $z \oplus y$. So we need also 4 transmissions. We can notice easily that using this method will produce some redundant transmissions as A and B receive z twice. In order to avoid these redundant transmissions we propose to integrate the network coding with MPR technique, then only the nodes selected as MPR are allowed to perform the coding and rebroadcast the packets. So in the example when C and M receive the two packets x, y only M will rebroadcast $x \oplus y$ as it is MPR for both A and B. Thus 3 transmissions only are sufficient to deliver the two packets x, y to the entire network. So using MPR-based coding reduce the number of transmissions needed to broadcast these two packets.

The idea of MPR-based coding is that when a packet is received by node M, first it decodes the packet if it has been encoded, and then it inserts the new packet into a queue for rebroadcasting if and only if the packet was received from a node which has selected node M as its MPR. We call this node a MPR selector of M. As we show in detail later, MPR-based coding not only reduce the number of transmissions but also reduce the number of packets in the queue that each node keeps to rebroadcast. In our approach, as in the example above, we consider a simple opportunistic coding scheme which reduces the complexity of encoding and decoding functions. The opportunistic coding was used for unicast traffic in [8]. In this paper we show how this scheme could efficiently optimize flooding in wireless network.

The challenges that we need to address in this scheme are:

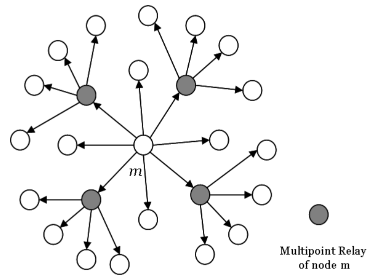


Figure 2: Diffusion of a broadcast message using multipoint relays

(a) Which packets should be encoded together? As each MPR node could has many packets to broadcast, it should find the set of packets to encode in order to maximize the number of packets delivered in a single transmission.To do that we need an efficient heuristic that optimizes the packet delivery and the waiting time.
(b) How can a node know if its neighbor can decode a packet? The node needs to know the state of neighbors, which is important for two reasons: First, for the encoding algorithm to efficiently choose the packets. Second, for stopping to send certain packets when they are delivered to all its neighbors.

3 MPR-BASED CODING PROTOCOL

We introduce a new protocol for ad hoc wireless network which integrates the MPR technique with the notion of network coding to achieve better performance while minimizing the packet delay, memory requirement and computation complexity.

3.1 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 [14]. So we propose to use the heuristic mentioned in [15].

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 following 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.

3.2 Network Coding Operation

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.

3.2.1 Problem formulation

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.

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

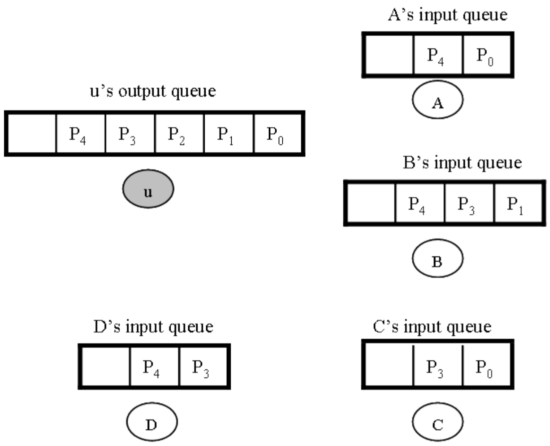
Now consider a node u which has N packets in its output queue and has M neighbors. We refer to the output queue by Q.

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

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.



(a) u want to find the coding candidate

the encoded packet	Number of neighbours that can decode
P0	2
P0+P1	3
P0+P2	2
P0+P3	3
P0+P4	3
P0+P1+P2	0
P0+P1+P3	2
P0+P1+P4	2
P0+P2+P3	1
P0+P2+P4	1
P0+P3+P4	4
P0+P1+P2+P3	0
P0+P1+P2+P4	0
P0+P1+P3+P4	1
P0+P2+P3+P4	0

(b) possible coding option

Figure 3: Exemple of coding operation

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. So we propose the following heuristic.

3.2.2 Distributed Packet Coding Algorithm

Our heuristic works as follow :

Step.0 Add the first packet of the output queue P_0 to the coding list.

Step.i Find the packet PP_i from the output queue which gives the largest increase in the number of neighbors that can decode the coding list after adding PP_i to it.

- If we find such packet we add it to the list, then repeat step_i if the number of packets in the list is inferior to M.
- Otherwise, If we don't find such packet or we have M packet in the coding list, then we suppose that $K = i$, XOR the packets in the list, add their IDs in the header and broadcast the encoded packets.

In this way, we reduce the number of cases that we have to test to $N \times M$. Lets take the example shown in Figure 3. Node u has 5 packets in its output queue. Some of these packets have been received by the neighbors of u as shown in Figure 3 (a). When u has an opportunity to send a packet, it picks P_0 from its output queue and try to find the other coding candidates. To do this, it has many possible combinations shown in the table in Figure 3 (b) we also add in the table the benefit of each combination represented by the number of neighbors that can decode it. We assume that u knows which packets each neighbor has. From the table we see that the best solution is $P_0 + P_3 + P_4$ and to get it we have to examine 15 possible combinations. However our heuristic will give us the result $P_0 + P_1$ after examining 8 cases only, which are the field shaded in 3. Therefore, we have lost one packet to deliver while reducing the number of test to the half. Another important point is the delay, we notice that our heuristic chooses to send P_1 instead of P_2 or P_3 since it has longer delay than the others. Therefore, even if our heuristic doesn't always give the best result, it gives a satisfactory result while at the same time it reduces significantly the complexity and the delay.

3.2.3 Learning Neighbor State

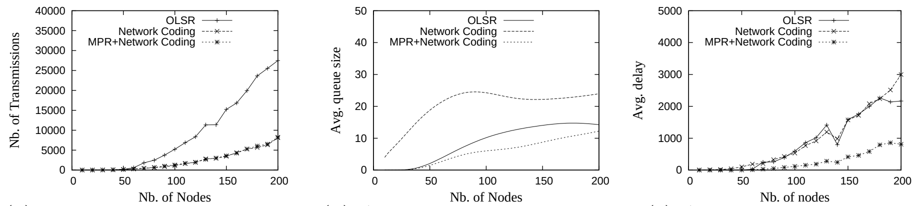
As we have seen in the coding algorithm, the key point to choose good coding candidates is to know the packets which exist at each neighbor. As explained earlier, we propose to associate each packet in the input queue with neighbor reception information. These information are updated according to the 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. For example, if node x receives a packet p from node y, node x can infer that node y has already p and that neighbors of y has received p. In the case where packet p is a coded packet, x can infer that y has all the coding candidate used to produce p. x can estimate the ability of each neighbor of y to decode p using the other neighbor reception information.

3.3 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. As mentioned earlier, the IDs of the coding candidates are listed in the header of the encoded packet. Moreover, all the received packets are stored in the input queue. So when a node receives an encoded packet, it checks if it can decode it depending on the decoding rule mentioned in section 3.2.1. If this is possible, it decodes the packet by simply XORing it with the known coding candidates stored in the input queue and get a new native packet. Finally, it schedules an acknowledgment to inform its neighbors about the new native packet that it has received.

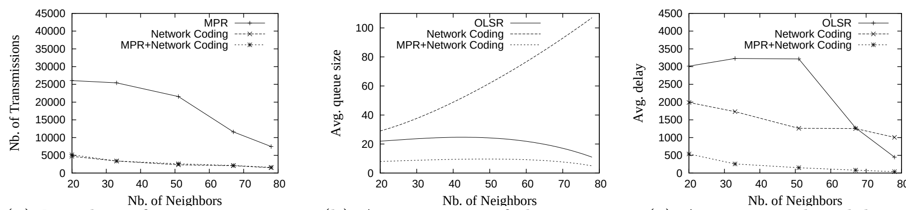
4 Simulation Results

To evaluate the efficiency of our MPR-based coding protocol for the flooding, we compare its performance with two other protocols; one is OLSR which optimizes the flooding by using MPR technique only. The other protocol is the one which use only the network coding without using the MPR technique. For this second protocol we use our algorithm for coding at each node in the network not only for the MPR nodes. Our performance metrics are the number of transmissions needed for flooding, memory requirement and the packets delay. To evaluate the memory requirement of the three protocols we calculate the average size of the output queue. At this queue a node keeps the packets that should be rebroadcasted until MAC indicates a send opportunity. And for the evaluation of the packets delay we use the average waiting time of the packets in this queue at all the nodes. 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 increase 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) Number of transmissions (b) Average size of the queue (c) Average packet delay

Figure 4: performance comparison of MPR-based coding and OLSR and only-network-coding-based flooding.



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

Figure 5: Impact of Node Density.

4.1 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. The number of nodes changes from 30 to 200. As shown in figure 4(a) that our MPR-based coding protocol reduces significantly the number of transmissions over the OLSR. 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 4(b)&(c) the MPR-based coding reduces the memory requirement and the packet delay over the protocol that uses network coding only because restricting the MPR nodes only to perform the coding reduces the number of packets in the queues. In the other hand, 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 over the OLSR.

4.2 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. 5 which shows that increasing the node density doesn't have great impact on the performance of MPR-based coding, instead it performs well for different node density. In contrast, OLSR performs better in dense network as there will be fewer nodes to broadcast and thus reduce the number of transmissions and the rate of draining packets from the queue which reduces the queue size and the packet delay as well. As shown in Fig. 5(b) that MPR-based coding reduces greatly the size of the queue over the case where only network coding is employed especially for dense networks. This is because increasing the number of neighbors increases the number of packets received by each node. However in the second case all of these received packets will be inserted in the output queue to be broadcast and this increases significantly the size of the queue, packet delay and also the complexity of the coding algorithm. So we deduce that our MPR-based coding protocol outperform the other tow protocols for different node density.

5 Summary and Conclusions

We have shown that incorporating network coding with the notion of multipoint relay allows to improve the broadcast efficiency over ad hoc wireless network. We developed simple distributed coding algorithm that allows to reduce the number of transmissions compared to OLSR which use only MPR technique and also reduce the memory requirement and the packet delay comparing with another protocol that use only the notion of network coding. Our simulation results demonstrated these benefits.

This work is a first step towards a full communication architecture. In the future, we intend to investigate a number of issues such as MPR-based coding 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)," 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] Ahlswede, Cai, Li, and Yeung, "Network information flow," *IEEETIT: IEEE Transactions on Information Theory*, vol. 46, 2000.

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

[6] 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, Aug. 2004.

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

[8] 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.

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

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

[11] 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. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2007.203

[12] 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. 05, August 2005.

[13] 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.

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

[15] 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.