Network Coding in Ad Hoc Networks: a Realistic Simulation Study

Saed Tarapiah, Claudio Casetti, Carla-Fabiana Chiasserini
Dipartimento di Elettronica, Politecnico di Torino

I. INTRODUCTION

In ad hoc networks, broadcasting is used for data dissemination with several different aims such as finding a route to a particular node, sending a warning signal or performing service discovery. The simplest method to implement broadcasting is flooding, though it can lead to undesired effects such as the *broadcast storm problem*: redundant packet transmissions resulting in repeated contention, collisions, and extra-power consumption. Typical solutions consist in curbing redundant transmissions and differentiating the timing of retransmissions.

In our study we address the problem of efficiently supporting broadcast traffic in ad hoc networks by using network coding. This technique can indeed reduce the overhead due to multiple copies of broadcast transmissions, by letting intermediate nodes encode multiple packets into a single output packet. Several works, e.g., [1], have shown that network coding can provide high reliability and reduced delays, beside improving traffic distribution and bandwidth use. However, most of previous works have shown these benefits either via theoretical analysis or in simplified simulation scenarios.

Here, we evaluate the impact of *random linear network coding*, in a realistic ad hoc network environment through the network simulator ns-2. Using a realistic MAC layer, we observe the effects of packets loss and propagation delay on network coding. Next, we compare the performance of broadcasting based on network coding with that of two simpler schemes, namely flooding-based and deferred broadcasting.

II. RANDOM LINEAR NETWORK CODING: AN OVERVIEW

Below, we briefly introduce the notations we use and the three main operations in random linear network coding.

Encoding. Let $M^1, ..., M^n$ denote the source native packets, and s denote the generation size, i.e., the maximum number of native packets that can be encoded together in one new encoded packet. Note that the header of each encoded packet must specify the generation to which the packet belongs. An encoding node outputs a generic encoded packet: $Y_j = \sum_{i=1}^{s} e_{j,i} M^{j+i-1}$ with $e_{j,i} \in GF(2)$, where $\mathbf{e_j} = (e_{j,1}, e_{j,2}, ...e_{j,s})$ denotes the coding vector. The coding vector is included in the header of the transmitted packet and it is used by receivers to decode or to re-encode the data.

Re-encoding. When a node has received $u \leq s$ encoded packets belonging to the same generation, then the node may generate a new encoded packet, by picking a new random vector $\mathbf{g_k} = (g_{k,1}, g_{k,2}, ...g_{k,u})$, and by computing the linear combination $X_k = \sum_{i=1}^{u} g_{k,i} Y_{k+i-1}$ with $g_{k,i} \in GF(2)$. It must be noted that the coding vector with respect to the original packets $M^{k+i-1}, ...M^{k+s-1}$ is now $\mathbf{e'_k} = (e'_{k,1}, e'_{k,2}, ...e'_{k,s})$, where $e'_{k,i}$ is given by: $e'_{k,i} = \sum_{h=1}^{u} g_{k,h} e_{h,i}$ The vector $\mathbf{e'_k}$ is included in the newly encoded packet header.

Decoding. Decoding at any receiver is performed by collecting packets of a given generation. These packets yield a system of linear equations that need to be solved to retrieve the original native packets. Suppose a node has received v encoded packets $X_1, X_2, ...X_v$ belonging to a given generation, with $v \leq s$ while $\mathbf{e'_1}, ...\mathbf{e'_v}$ represent the coding vectors corresponding to the encoded packets. The generic element of the decoding matrix G is given by: $G_{ij} = e'_{ij}$ where $i = 1, \dots v$ and $j = 1, \dots s$. Let us denote the rank of G by R. When the matrix has full rank, i.e., $R = v = s$, for a given generation, then the node can solve the linear equations to retrieve all native packets belonging to that generation. By deriving $e'_{l,i}$ from the packet headers and solving the linear equations with M^i as the unknowns, the destination can recover the set of native packets $M^l, ...M^{l+v}$. The receiver can also perform an early decoding when a G sub-matrix has full rank. In this case, the receiver can recover part of the source native packets belonging to the given generation. We finally observe that when a node receives a packet, it must check whether it is *innovative* or not, i.e., whether it increases the rank of the decoding matrix G. If not, the packet is dropped.

III. BROADCASTING SOLUTIONS

Here we introduce our network coding-based scheme and the two other broadcasting schemes that provide a term of comparison with our own.

Network Coding-based Broadcasting. We detail the different operations that, according to our scheme, are performed by the network nodes. Note that each node (except for the source) needs to collect independent packets for re-encoding/decoding, therefore some buffer space is needed at the receiver. We will call this buffer *NC buffer*.

Source node operations. The source's application layer generates native packets, which are then encoded at the network coding sub-layer. The source collects s native packets and encodes them to generate s different encoded packets $Y_1, ...Y_s$, as explained in Section II.

978-1-4244-3968-3/09/$25.00 ©2009

Intermediate node operations. When an intermediate node receives the first encoded packet of a generation, the packet is cached in the NC buffer and a timer is started. The node then has to establish whether the subsequently received encoded packets belonging to the same generation are innovative. The innovative packets are cashed, while the others are dropped. When G has full rank, the node performs decoding to retrieve the native packets belonging to the given generation. The native packets are then encoded again into one packet with a random independent coding vector, so that only one packet is forwarded. If the timer of the NC buffer associated to that generation expires before the matrix rank reaches s, only the set of encoded packets buffered at the node is re-encoded before a new encoded packet is forwarded.

Receiver node operations. Upon receiving encoded packets, the innovative ones are cached for decoding. After building G, its rank is checked. If G has full rank, the receiver recovers all native packets in a given generation; an early decoding is performed when a sub-matrix of G has full rank.

Simple Flooding and Deferred Broadcast. The flooding scheme we use as term of comparison is based on the IEEE 802.11 standard, according to which broadcast transmissions are never acknowledged. To reduce the collision probability, we let a node receiving a packet defer its retransmission by a random time, chosen in the range [0, 10] ms.

The deferred broadcast, instead, includes the mechanism proposed in [2]. Before rebroadcasting a packet, a node waits for an hold-off time during which it listens to the channel: if it hears the same packet being broadcast by two other nodes with different predecessors, it drops the packet, otherwise it rebroadcast it. Being the hold-off time computed according to the received signal strength, nodes that are farther away from the sender compute a shorter delay and rebroadcast first.

IV. PERFORMANCE EVALUATION

To assess and compare the performance of the broadcasting schemes just outlined, we use the network simulator ns-2. The propagation on the physical channel is simulated using the two-ray ground model and a data rate of 1 Mb/s is assumed for the 802.11 protocol. The application layer at the source node generates 1000-byte long CBR packets. We consider static, random topologies including 100 nodes uniformly deployed over an area of 100×100 m. The performance metrics are averaged over 10 instances of random network topologies.

The metrics used to evaluate the broadcast schemes are: delivery delay, packet loss rate, transmission fairness and protocol overhead at the MAC layer. The latter metric tracks the total number of bytes needed to broadcast a packet; hence, it is related to the energy consumed by broadcasting across the whole network.

We study the system performance as the average number of neighbors per node varies, i.e., as the node radio range varies, and for different traffic rates. In addition, for our network coding-based broadcasting, we consider different values of generation size ($s = 2, 3, 4, 5$) and of maximum buffering timeout of the NC buffer. Below, we briefly summarize the main results in terms of packet loss rate, transmission fairness and protocol overhead, obtained for a traffic rate and a maximum NC buffer timeout equal to 50 kb/s and 1 s, respectively.

Packet loss: As the network becomes more and more connected, nodes can receive packets from different neighbors thus having more chance to receive all broadcast packets. Hence, fewer packets are lost in denser neighborhoods. When compared to network coding, flooding and deferred broadcast lose more packets, but only for a neighborhood size greater than 12 and $s \leq 3$. Indeed, the random node deployment may lead to very dense neighborhood where the medium sharing conditions become extremely crowded, resulting in higher collision probability at the MAC layer. We also remark that our findings differ from previous work showing that network coding-based broadcasting outperforms flooding in dense networks.

Protocol overhead: In the case of flooding, the larger the neighborhood size, the larger the number of transmissions and, thus, the overhead. When, instead, the deferred broadcast technique is used, the opposite behavior is observed. Indeed, when the radio range is shorter, the network is scarcely connected and only few nodes can hear each other; it follows that several nodes must re-broadcast traffic packets. As the radio range increases, many nodes refrain from re-broadcasting the packets as they hear their neighbors transmitting first. Network coding-based broadcast provides similar results to those achieved by deferred broadcast, although its performance significantly depends on the generation size: the larger the generation size, the more packets are encoded into one, and the smaller the broadcast overhead.

Fairness: By computing the well-known Jain's fairness index among all nodes, we observe that the flooding and network coding schemes provide high index values for neighborhood size greater than 4, while worse fairness is recorded for smaller values due to some unconnected nodes. Conversely, deferred broadcast has poor fairness, since the same set of nodes is selected as forwarder for all traffic packets, namely the nodes that ensure the greater spatial progress for the packets.

On-going work: In our current work on the network coding scheme, we derive results as the packet size and the number of encoding operations made by intermediate nodes vary. Also, we track the number of packets used for re-encoding by intermediate nodes as well as the percentage of innovative packets that a node receives. Finally, we assess the performance considering nodes mobility and a more sophisticated channel propagation model.

ACKNOWLEDGEMENT

This work was supported by Regione Piemonte through the VICSUM project.

REFERENCES

[1] C. Fragouli, J. Widmer, J.-I. Le Boudec, "Efficient Broadcasting Using Network Coding," *IEEE/ACM Trans. on Networking*, vol. 16, no. 2, 2008.

[2] V. Naumov, R. Baumann, T. Gross, "An Evaluation of Inter-Vehicle Ad Hoc Networks Based on Realistic Vehicular Traces," *ACM MobiHoc*, 2006.