Revisiting XOR-based Network Coding for Energy Efficient Broadcasting in Mobile Ad Hoc Networks

Nikolaos Papanikos[a], Evangelos Papapetrou[a,]

[a]*Department of Computer Science and Engineering, University of Ioannina, GR 45110, Greece*

Abstract

Network coding is commonly used to improve the energy efficiency of network-wide broadcasting in wireless multi-hop networks. In this work, we focus on XOR-based broadcasting in mobile ad hoc networks with multiple sources. We make the observation that the common approach, which is to benefit from the synergy of XOR network coding with a CDS-based broadcast algorithm, suffers performance breakdowns. After delving into the details of this synergy, we attribute this behavior to an important mechanism of the underlying broadcast algorithm, known as the "termination criterion". To tackle the problem, we propose a termination criterion that is fully compatible with XOR coding. In addition to that, we revisit the internals of XOR coding. We first enhance the synergy of XOR coding with the underlying broadcast algorithm by allowing each mechanism to benefit from information available by the other. In this way, we manage to improve the pruning efficiency of the CDS-based algorithm while at the same time we come up with a method for detecting coding opportunities that has minimal storage and processing requirements compared to current approaches. Then, for the first time, we use XOR coding as a mechanism not only for enhancing energy efficiency but also for reducing the end-to-end-delay. We validate the effectiveness of our proposed algorithm through extensive simulations on a diverse set of scenarios.

Keywords: energy-efficient broadcasting, network coding, mobile ad hoc network, XOR-based coding, connected dominating set, deterministic broadcasting

1. Introduction

Network-layer broadcasting is fundamental for mobile ad hoc networks because it provides the means to disseminate information throughout the network. Besides application data, broadcast protocols are also used to distribute control information to every network node [1, 2]. In this way, each node maintains a view of the network structure that is a basic element for many networking mechanisms. The seminal work by Ahlswede et al. [3] introduced network coding, a concept that significantly enhances the performance of networking protocols in both wired and wireless networks. As a result, over the last years, many researchers focused on incorporating network coding into broadcasting in wireless ad hoc networks [4–25]. Based on their main objective, coding-based approaches can be classified into: i) energy-efficient, and ii) delivery guarantee schemes. The schemes of the first class [4–13] utilize network coding towards energy efficiency aiming to strike the best possible balance between delivery and cost (as expressed by the number of transmissions). On the other hand, the schemes of the second class [14–25] use network coding to guarantee the delivery of broadcast packets to all network nodes, treating the minimization of the related costs as a secondary objective.

In this work we focus on energy-efficient broadcasting in mobile ad hoc networks. Moreover, we are interested in the scenario of multiple broadcasting sources, i.e. we examine the many-to-all and all-to-all communication paradigms. Such scenarios appear rather frequently when multiple nodes in parallel and independently engage in discovery phases. Some representative examples include discovering routes in on-demand routing protocols [1], locating resources in service discovery applications [2, 26] and retrieving volatile data from peer databases [27, 28]. In all of the aforementioned examples the focus is on energy efficiency rather than on guaranteeing delivery.

In the field of energy efficient broadcasting the most popular design choice is to adopt XOR-based network coding [29]. Algorithms that follow this approach [4–9] encode packets on a hop-by-hop basis using bitwise XOR and then forward them using a Connected Dominating Set (CDS) based broadcasting scheme [30–33]. Although this strategy has been proved successful, we bring to light several occasions where its performance severely degrades and the coding gain becomes negligible. Motivated by this finding, we examine in depth the synergy of network coding and the underlying CDS-based broadcast algorithm. We conclude that the weak link is the mechanism of the broadcasting algorithm known as "the termination criterion". Therefore, as a first step, we explore the use of alternative termination criteria proposed in the literature of traditional broadcasting. Unfortunately, we find that none of them is compatible with XOR network coding. To address the problem, we propose the *Network Coding Broadcast with Coded Redundancy* (NOB-CR) algorithm which incorporates a novel termination criterion that is fully compatible with XOR coding.

Email addresses: `npapanik@cse.uoi.gr` (Nikolaos Papanikos), `epap@cse.uoi.gr` (Evangelos Papapetrou)

Moreover, NOB-CR revisits the coding internals in order to enhance the overall performance in terms of energy efficiency, delivery delay and utilization of network resources. In summary, our main contributions are:

- We unveil the shortcomings in the synergy between XOR coding and CDS-based broadcasting (Section 3). Then, after analysing the reasons of this finding, we propose a coding-friendly termination criterion for the CDS-based algorithm and illustrate its efficiency (Section 4).

- We delineate a novel method for detecting coding opportunities (Section 5.2). The method is lightweight in the sense that, in contrast to current approaches, requires each node to maintain minimum state while it mostly utilizes information that is already available through the underlying broadcast mechanism.

- We enhance the pruning efficiency of the underlying CDS-based algorithm by exploiting information available from the coding mechanism, i.e. we establish a bidirectional synergy between network coding and CDS-based forwarding (Section 5.3).

- We address the problem of increased end-to-end delay in network coding broadcasting (Section 5.4). This problem is a direct consequence of Random Assessment Delay (RAD), a mechanism used by XOR coding in an effort to increase coding opportunities. The proposed solution, called *Coded Redundancy*, takes a novel approach and uses XOR coding to achieve a cost-free increase of packet redundancy across the network in order to reduce end-to-end delay.

The rest of the paper is organized as follows. In Section 2, we review the basic principles of CDS-based broadcasting and XOR network coding. In Section 6, we present an extensive evaluation of the proposed algorithm. In Section 7, we review the existing work related to coding-based broadcasting in wireless ad hoc networks. Finally, we summarize our findings in Section 8.

2. Preliminaries

Before continuing, we first briefly review the basic principles of CDS based broadcasting as well as XOR-based coding.

2.1. CDS-based Broadcast Principles

Energy efficient broadcast algorithms aim to minimize the number of transmissions required for delivering a packet to all network nodes [34, 35]. The most effective algorithms follow the CDS-based broadcasting approach. According to this, the algorithm constructs a connected dominating set of the network [30–32]. The nodes constituting the CDS are the *forwarders*, i.e. those elected to forward the broadcast packets, while all other nodes just act as passive receivers. Since computing the forwarders should be performed in a distributed fashion, the common approach is to approximate them locally at

each node v using its 1-hop neighbor set ($\mathcal{N}(v)$), i.e. the set that consists of v's one hop neighbors, and the 2-hop neighbor set ($\mathcal{N}(\mathcal{N}(v))$), i.e. the set consisting of all nodes that lie at maximum two hops away from v. More specifically, forwarders are elected so as to deliver the packet to (or "cover") the set $\mathcal{U}(v)$ of nodes that lie exactly 2-hops away from v, i.e. $\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v)$. The set of candidate forwarders $C(v)$ consists of v's neighbors, i.e. $C(v) = \mathcal{N}(v)$. Note that $\mathcal{U}(v) \subseteq \bigcup_{\forall u \in C(v)} \mathcal{N}(u)$ and that $C(v)$ can be seen as a set of sets if each node $u \in C(v)$ is replaced by $\mathcal{N}(u)$, thus the election of forwarders is modeled as a set cover problem. The solution is usually given by the well-known greedy set cover (GSC) algorithm [36], however other more efficient approximation algorithms exist [31, 37–39].

Even though transmitting packets only through forwarders successfully reduces packet duplicates, a significant number of them still exists across the network. This is because the selection of forwarders is made in a distributed manner and with limited information. As a result, special attention should be given to these duplicates as they could lead to additional transmissions and degrade energy efficiency. Therefore, the reception of a packet duplicate in a forwarder node leads to a dilemma whether to forward it or not. Forwarding the duplicate could increase redundant transmissions while dropping it could potentially impact the delivery efficiency. The mechanism that is responsible to handle such situations is the *termination criterion*. Multiple criteria have been proposed in the literature [33, 40–42]. The most well-known approach requires that each node keeps track of the packets received by each of its 1-hop neighbors. Then, in the case of a duplicate reception, a forwarder transmits it if at least one of the neighbors is not marked to have received the packet. Having all nodes to store the reception status for all of their 1-hop neighbors and for all packets could be a daunting challenge in terms of both memory usage and processing overhead [33]. In the rest of this paper, we refer to this approach as the *marked/unmarked (M/U)* [33] criterion. Another termination criterion mandates that a forwarder transmits a duplicate only if no other duplicate of the same packet has been relayed by the same forwarder in the past. We refer to this scheme as the *relayed/unrelayed (R/U)* [33] criterion. The last termination criterion is based on the following simple rule. A node acting as a forwarder relays packets seen for the first time while it drops already seen packets including the ones not relayed in the past because the node was not elected as a forwarder at that time. We refer to this approach as the *covered/uncovered (C/U)* [40–42] criterion. In contrast to the M/U criterion, the latter two approaches are more realistic due to the limited storage and processing requirements.

The algorithms proposed in the literature follow two major strategies for building the CDS, i.e. calculating the forwarders. The first is to build a CDS for the entire network while the second is to build a source-specific CDS. MPR [31] and DP [32] are representative algorithms of the two approaches. In MPR, the forwarders, called multipoint relays, are computed by modeling the problem as a set cover one. The multipoint relays are used for any packet regardless of its source and updated whenever topology changes are detected. On the other hand, in

DP, a node that relays a packet calculates the list of forwarders by considering the previous hop of the packet and piggybacks the corresponding list on it. In this way, a source-based CDS is formed for each packet. More specifically, when a node v receives a packet from u checks whether it is selected as a forwarder. If so, it takes advantage of u's neighborhood to reduce both the set of candidate forwarders and the number of nodes that should receive the packet, i.e. $C(v) = \mathcal{N}(v) - \mathcal{N}(u)$ and $\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v) - \mathcal{N}(u)$ respectively. Other algorithms in the sourced-based CDS category vary in the approach taken to minimize the set $\mathcal{U}(v)$ and therefore the number of forwarders. TDP and PDP [33] exploit u's two-hop neighborhood and further minimize the $\mathcal{U}(v)$ set. For example, node v in PDP elects forwarders in order to cover the nodes in $\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v) - \mathcal{N}(u) - \mathcal{N}(\mathcal{N}(u) \cap \mathcal{N}(v))$. In the context of this paper, for presentation purposes we selected Partial Dominant Pruning (PDP) [33] as the reference algorithm. However, our findings can be easily generalized to all CDS-based broadcast protocols. The reason is that, as far as the CDS-based operation is concerned, we focus on the termination criterion which is a generic mechanism that does not depend on the specifics of each algorithm.

2.2. XOR Coding Specifics

XOR-based coding works on a hop-by-hop basis, i.e. packets encoded by a node are decoded by its neighbors. The idea is that each node v can combine packets using bitwise XOR operations in order to produce an encoded packet. For the neighboring nodes to be able to decode the encoded packet, the choice of native, i.e. non coded, packets is important. More specifically, for a successful coding of k packets , each neighbor should know $k-1$ of those packets beforehand. This requirement guarantees that each neighbor should be able to decode the encoded packet. The existence of $k > 1$ packets that can be encoded is known as a *coding opportunity* [29]. It is clear that, finding a coding opportunity depends on v's knowledge about the packets that each of its neighbors has already received. To acquire such information, v employs opportunistic listening [4, 29] and snoops all communication in the wireless medium. The acquired information is stored in what is called the *neighbor reception table*. Moreover, node v should store in what is called the *packet pool* all recently received native packets in order to be able to perform decoding of encoded packets. To describe the method more formally, let $\mathcal{P}_v$ denote v's packet pool, i.e. the set of native packets recently received by v and $\mathcal{R}_v^u$ denote u's view of the same buffer. Note that $\mathcal{R}_v^u$ is part of u's neighbor reception table. Node v may choose a set of native packets $\mathcal{B}' \subseteq \mathcal{P}_v$ and produce an encoded packet, by using bitwise XOR, in the presence of a coding opportunity. This means that a set $\mathcal{B}' \neq \emptyset, |\mathcal{B}'| > 1$ can be found such that, according to v's neighbor reception table, each node $u \in \mathcal{N}(v)$ has received at least $|\mathcal{B}'| - 1$ of the native packets in $\mathcal{B}'$, i.e. $|\mathcal{R}_u^v \cap \mathcal{B}'| \geq |\mathcal{B}'| - 1, \forall u \in \mathcal{N}(v)$. Successful decoding depends on the consistency of $\mathcal{R}_u^v$, i.e., whether $\mathcal{R}_u^v \subseteq \mathcal{P}_u$. Decoding failures at a node u occur when $|\mathcal{P}_u \cap \mathcal{B}'| < |\mathcal{B}'| - 1$ and result in the loss of all packets included in the encoded one.

The efficiency of XOR-based coding clearly depends on the existence of coding opportunities. This is because for each encoded packet that contains k native ones only one transmission is required instead of k, thus saving energy and reducing packet collisions. To maximize the number of coding opportunities XOR-based coding approaches introduce a *Random Assessment Delay (RAD)* before relaying a packet. Higher values of RAD result in more candidate packets for encoding, however this comes at the cost of increased end-to-end delay.

3. The synergy between network coding and the termination criterion

XOR network coding as well as the termination criterion of a CDS-based algorithm are essential mechanisms for energy efficient broadcasting as both aim to minimize packet transmissions. Ensuring a smooth synergy is critical for building an efficient algorithm. Most proposed coding-based broadcast algorithms combine XOR-coding with CDS-based approaches that utilize the M/U criterion [4–7] while others do not provide insight on the termination criterion used [7, 8]. In the following we will show that M/U faces performance issues that hamper the coding operation. At the same time, using other proposed termination criteria, such as R/U and C/U, in parallel with XOR coding raises significant design issues.

3.1. The M/U criterion limits coding gains

The choice to combine M/U with XOR coding is reasonable. First, M/U is compatible with the RAD technique that is essential for network coding. In fact, RAD improves the pruning efficiency of M/U. This is because the imposed delay allows the reception of more packet duplicates which could potentially change the initial decision to relay the packet. This is the reason for which RAD has also been proposed in the context of broadcasting without network coding [34, 35]. However, although in non-coded approaches there are alternatives that provide performance improvements similar to that of RAD but without the associated delay, in coding-based approaches using RAD is essential. The second reason for which state-of-the-art XOR-based approaches adopt the M/U criterion is because in this case XOR coding can be implemented with limited cost. Recall that the latter requires information about the reception status of the neighboring nodes, i.e. the neighbor reception table. This is exactly the information on which M/U decisions are based on, therefore this information is already available through the implementation of M/U. Besides the advantages of using M/U there is also a significant downside. M/U is known to be less efficient than other proposed criteria [40, 41]. This motivated us to further examine the performance of XOR-based broadcasting implementing M/U against non coding schemes utilizing the other termination criteria, i.e. R/U and C/U.

For our investigation, we conducted a series of experiments using the ns2 simulator [43]. We chose to experiment with the well-established CodeB algorithm [4] that utilizes network coding and builds on top of PDP using the M/U termination criterion. In our experimental setup, 100 nodes move with maximum speed of 1 m/sec in a square area according to the Random Waypoint (RW) model [44]. Each node has a neighborhood with an average size of 15 nodes, while 50 nodes generate
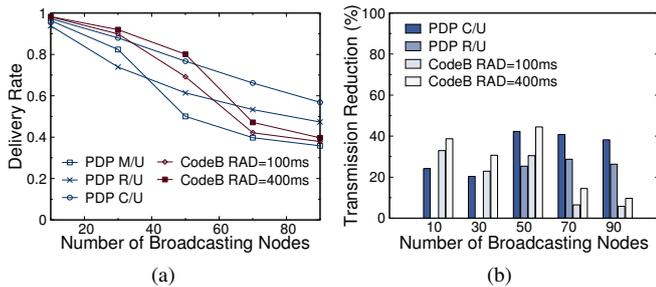
Figure 1: Performance under varying traffic load: (a) Delivery rate (b) Transmission reduction compared to PDP M/U
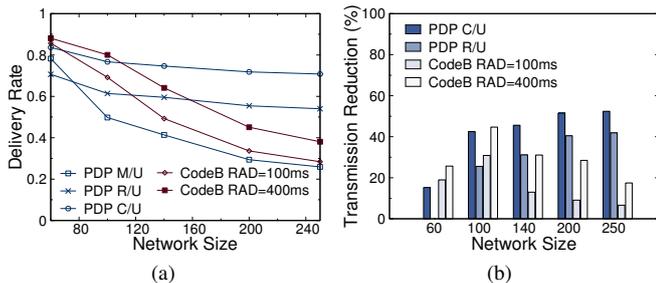

Figure 2: Performance for different network sizes: (a) Delivery rate (b) Transmission reduction compared to PDP M/U

broadcast traffic with a rate of 1 packet/sec. More information about the simulation set-up can be found in Section 6. First, we evaluated the performance of CodeB under various levels of offered traffic by varying the number of source nodes. Fig. 1a depicts its delivery efficiency, while Fig. 1b displays its ability to prune transmissions compared to the non-coding PDP scheme that uses the M/U criterion. As expected, CodeB successfully reduces the number of transmissions and provides enhanced packet delivery in cases of low to medium traffic load. However, as the network traffic increases its efficiency deteriorates. Based on this observation, we attempted to boost CodeB's performance. More specifically, we included a new version of CodeB that maximizes the benefits of network coding by increasing the RAD value from 200 ms to 400 ms. However, despite the fact that the new CodeB version clearly discovers more coding opportunities and thus further reduces transmissions (Fig. 1b), its improvement in terms of delivery efficiency is limited (Fig. 1a). To further investigate the reasons behind CodeB's poor behavior, we compared it against two versions of the non-coding PDP scheme, one implementing the R/U criterion and the other the C/U. If there exists at least one simple PDP scheme that performs better than CodeB then the origins of the witnessed poor behavior reside in the termination criterion rather than the coding mechanism itself. Interestingly, this is confirmed by our results in Fig. 1. After the breaking point of 50 broadcast sources (half of the network nodes), the non-coding PDP schemes outperform CodeB regardless of the RAD value used. The only exception is PDP M/U that, similar to CodeB, suffers from a performance breakdown.

Similar findings are witnessed in our second experiment where we assess the scalability of all algorithms by increasing the number of network participants (Fig. 2). CodeB outperforms all schemes for networks comprised of fewer than 100

participants. Despite the fact that the offered load remains constant as the network size increases, CodeB and PDP M/U cannot avoid performance breakdown (Fig. 2a). Both generate a large number of transmissions (Fig. 2b) that induce failures due to packet collisions. Increasing the RAD value offers CodeB a performance improvement, however the gain is still limited and the problem is not solved. On the other hand, the non-coding schemes, PDP R/U and C/U, present a relatively stable behavior regardless of the network size. Clearly, the best performing algorithm is PDP C/U that reduces transmissions by up to ~60% while keeping the delivery efficiency above ~75%.

Overall, the results revealed that the combination of network coding with the M/U termination criterion is not always the best choice. In particular, non-coding schemes perform far better than CodeB when the traffic in the network increases; either because more traffic is offered from more sources (first experiment) or because in a bigger network (second experiment) more forwarders exist and produce more packet duplicates. As explained, this behavior is not the result of the coding operation itself but is inherited from the underlying broadcast scheme and more specifically the termination criterion. There are two reasons for this. The first and predominant one is the limited ability of M/U to prune redundant transmissions. As a result, congestion quickly builds up and results in more collisions, therefore reducing delivery efficiency. The second reason is related to the neighbor reception table, i.e. the structure containing information about the packets received by each neighbor, which is necessary for both M/U and XOR coding. In the typical implementation of this structure, information for each packet is maintained for a limited time period. As traffic in the network increases, the delay jitter between the first and the last duplicate of a packet also increases. As a result, there is an increased probability that a packet duplicate arrives at a node v after the information for that packet has expired and been removed from the neighbor reception table. This results in node v transmitting more duplicates and thus aggravating congestion. Increasing the expiration period for information in the neighbor reception table improves performance up to a limit. After that, no further improvement is possible and performance breakdown is still evident due to the limited pruning ability of M/U. We even implemented the neighbor reception table as a fixed size structure without imposing an expiration period. We tested different sizes and found that performance degradation appears to be more severe in this case.

3.2. The pitfalls of using other termination criteria

Our observations highlight the need for replacing M/U with alternative criteria, such as R/U and C/U. However, doing so is not straightforward. The main reason is what we call "the packet reordering problem". Before analyzing the packet reordering problem, let us first describe the implementation aspects of the two alternative termination criteria. Recall that both R/U and C/U delineate a policy for handling packet duplicates. More specifically, in R/U a forwarder relays a duplicate only if no other duplicate of the same packet was relayed in the past. On the other hand, in C/U a forwarder v relays only the packets seen for the first time and ignores packets seen in the past
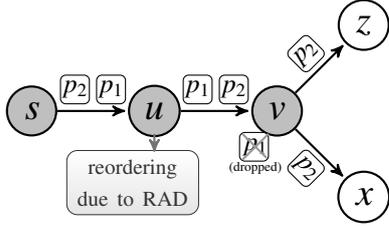
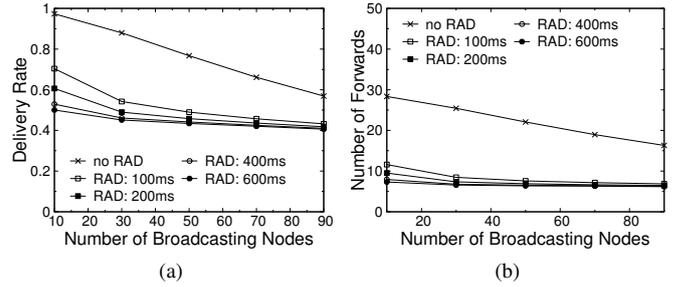Figure 3: Example where the propagation of packet p_1 terminates due to the packet reordering problem

Figure 4: Performance of PDP C/U vs traffic load using different RAD intervals: (a) Delivery efficiency (b) Average number of forwards per packet

even if v did not forward those packets, i.e. v was not selected as a forwarder at that time. In order for both R/U and C/U to function properly, there are two prerequisites. The first is that packets should be uniquely identified through a number added by the source node at creation time. The second prerequisite is that each node implementing the termination criterion should store a full reception history on a packet basis (i.e. the id's of received packets). This is neither practical nor realistic due to the high storage and processing requirements. For this reason, the traditional implementation of both R/U and C/U takes a much simpler approach. The numbers used to identify packets are assigned by the source in a sequential manner (thus called sequence numbers) so as packets with higher numbers correspond to the ones created more recently. This allows each node v that implements either R/U or C/U to only store a single sequence number (SN_s) for every source node s. In the R/U criterion (C/U criterion), this is the largest number seen in a packet from s and forwarded (received) by v. Then, for an incoming packet p_1 carrying the sequence number SN_{p_1} it is sufficient to check that $SN_{p_1} > SN_s$ so as to decide that it is not a duplicate.

Unfortunately, the aforementioned implementation is fully functional only under the assumption that all nodes in the network receive packets in the same order in which they were created. When this order is altered the problem that we call *packet reordering* emerges, impairing the ability of both R/U and C/U to detect duplicates and therefore having a severe impact on their performance. The utilization of XOR coding unfortunately results in packet reordering and thus its incompatibility with current implementations of both R/U and C/U. More specifically, packet reordering appears due to the random assessment delay (RAD) that network coding uses at each node in order to maximize the probability of finding a coding opportunity. To make it more clear, let us examine the problem through an example in which both R/U and C/U fail to work properly. Fig. 3 illustrates the propagation of packets p_1 and p_2 across an example network. Both packets originate from the same source s and p_1 is created before p_2. Therefore, the sequence number of p_1 is smaller than that of p_2, i.e., $SN_{p_1} < SN_{p_2}$. When RAD is not utilized, u will forward both packets in the same order as received. Then, v will first receive p_1, update SN_s, i.e. $SN_s \leftarrow SN_{p_1}$, and finally forward p_1. Upon reception of p_2, v will confirm that $SN_{p_2} > SN_s$ and will forward p_2. On the other hand, if RAD is utilized, u introduces a random delay before forwarding p_1 and p_2. Due to randomness, the delay for p_2 may be significantly smaller than the corresponding delay for p_1, thus resulting in u forwarding the two packets in the reverse

order, i.e. p_2 first and then p_1. After receiving p_2, v updates SN_s to the value SN_{p_2} and forwards p_2. Later on, when v receives p_1 makes the observation that $SN_{p_1} < SN_s$, therefore rejects p_1 although it is not a duplicate. This decision has a major impact on the delivery efficiency since p_1 never reaches nodes z and x.

To validate the impact of the packet reordering problem we conducted a series of experiments on the PDP algorithm using the C/U termination criterion. We examined the effect of different values of the random assessment delay. For the experimental setup we used the same settings described in Section 3.1. Fig. 4 illustrates our main results. Clearly, RAD has a considerable impact on the overall broadcasting performance regardless of the offered load. More specifically, the packet reordering problem may result in a reduction of the number of relaying decisions up to ~3 times (Fig. 4b). However, this pruning is erroneous in the sense that it prematurely terminates the broadcasting process, thus, reducing the delivery efficiency up to ~50% (Fig. 4a). As expected, higher RAD values have a more severe impact on the performance as in these cases the probability of receiving packets out of order increases. We observed similar findings in the R/U case.

4. Building a coding friendly termination criterion

Establishing the compatibility of R/U and C/U with XOR coding requires solving the packet reordering problem. Allowing each node that implements either of these criteria to store a full packet reception history can provide a solution. However, as mentioned earlier, this approach is neither practical nor realistic due to the high storage and processing requirements. Towards a more efficient solution, we propose the *modified covered/uncovered (MC/U)* termination criterion that extends C/U. We choose to build on top of C/U because both the related literature [40, 41] and our experimental results (Fig. 1 and Fig. 2) confirm that it achieves the best performance against all other proposed termination criteria. The main idea behind our approach is to implement the same forwarding criteria as in C/U but to allow each node to store information (just one bit as we will discuss in the following) for each of the k, instead of just one, most recently seen packets from each source node. This allows the node to detect any duplicate of these k packets without problems caused by packet reordering. Duplicate detection is not possible for a packet that is older than the k recorded ones because no relevant information is available. However, this is
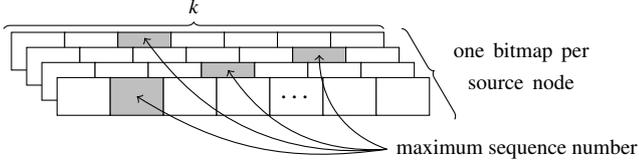
5

Figure 5: Structure used for the modified covered/uncovered (MC/U) termination criterion

important only if a copy of a packet p from source s is received by a node after the k-th packet that s generated after p. By increasing k it is possible to minimize the probability of such an occasion. Even if such an occasion arises we choose to drop the packet, i.e. adopt the C/U policy, rather than forwarding it (which corresponds to the M/U policy) in order to avoid increasing the network congestion levels.

Selecting a proper value for k is clearly a challenging task. Large values increase the storage and processing requirements at each node while small values increase the probability of receiving a packet without being able to decide whether it is a duplicate or not. After experimentation, we concluded that the MC/U criterion has a competitive performance even when a small value of k is required due to space limitations. Nonetheless, the storage and processing requirements may raise a concern. To address such concerns, we implement MC/U using bitmaps. Note that bitmaps have been used for similar purposes in the context of multicasting in ad hoc networks [45]. More specifically, a node v that implements MC/U, instead of storing the k last seen sequence numbers from a source s, it uses only one bit for each of them, i.e. a total of k bits in the form of a bitmap $\mathtt{BM_s}$ (Fig. 5). Then sequence numbers are mapped to the bits of $\mathtt{BM_s}$ and each bit is used to indicate whether the corresponding sequence number is known (bit set to 1), i.e. v has already received a packet carrying this sequence number, or not (bit set to 0). At the same time, by using bitmaps the node takes advantage of the low cost read/write operations. Furthermore, node v stores the maximum known sequence number from s

---

RelayOrNot(packet p, bitmap $\mathtt{BM_s}$, int $\mathtt{SN_s^{MAX}}$, int $\mathtt{mindex}$)

```
 1: if (p.SN > SN_s^MAX) then
 2:     Update(p,BM_s,SN_s^MAX,mindex)
 3:     relay p if forwarder
 4: else
 5:     SN_s^MIN ← SN_s^MAX − k
 6:     if (p.SN <= SN_s^MIN) then
 7:         drop p
 8:     else
 9:         index ← p.SN − SN_s^MAX + mindex
10:         if (index < 0) then
11:             index ← index + k
12:         end if
13:         val ← BM_s.get(index)
14:         if (val) then
15:             drop p
16:         else
17:             BM_s.set(index)
18:             relay p if forwarder
19:         end if
20:     end if
21: end if
```

Figure 6: Pseudocode of the MC/U forwarding procedure.

Update(packet p, bitmap $\mathtt{BM_s}$, int $\mathtt{SN_s^{MAX}}$, int $\mathtt{mindex}$)

```
 1: mindex′ ← (mindex + p.SN − SN_s^MAX)%k
 2: rollover ← ⌊ (mindex+p.SN−SN_s^MAX)/k ⌋
 3: if (rollover == 1) then
 4:     BM_s.zero(mindex + 1, k − 1)
 5:     BM_s.zero(0, mindex′ − 1)
 6: else if (rollover > 1) then
 7:     BM_s.zero(0, k − 1)
 8: else
 9:     BM_s.zero(mindex + 1, mindex′ − 1)
10: end if
11: mindex ← mindex′
12: SN_s^MAX ← p.SN
13: BM_s.set(mindex)
```

Figure 7: Pseudocode for updating the bitmap.

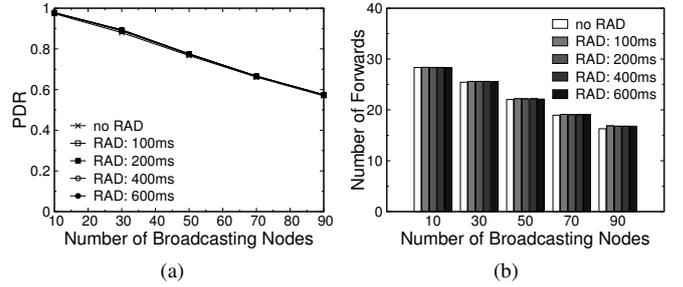Figure 8: Performance of PDP MC/U under varying traffic load using different RAD intervals: (a) Delivery efficiency (b) Average end-to-end delay

($\mathtt{SN_s^{MAX}}$) as well as the index ($\mathtt{mindex}$) of the bit in $\mathtt{BM_s}$ that corresponds to $\mathtt{SN_s^{MAX}}$.

The pseudocode of MC/U is illustrated in Fig. 6. When a node v receives a packet p from s it first checks whether its sequence number $p.\mathtt{SN}$ is greater than $\mathtt{SN_s^{MAX}}$. If this is the case v relays the packet (if it is an elected forwarder) and updates its state (Fig. 7). This update involves the following steps. First v calculates the index ($\mathtt{mindex′}$) of the bit that corresponds to the new sequence number (line 1). Observe that this calculation may involve a rollover, i.e. reusing the bits of the bitmap. Then v resets all the bits from position $\mathtt{mindex}+1$ to $\mathtt{mindex′}-1$ (lines 2-10). This is done because those bits correspond to the sequence numbers between $\mathtt{SN_s^{MAX}}$ and $p.\mathtt{SN}$ and no packet carrying one of these numbers has been received so far. Note that if $p.\mathtt{SN} - \mathtt{SN_s^{MAX}} > k$, i.e. a multiple rollover occurs, then all bits of the bitmap must be reset (lines 6-7). Finally, v updates $\mathtt{SN_s^{MAX}}$ and $\mathtt{mindex}$ (lines 11-12) and sets the corresponding bit to indicate that a packet carrying $\mathtt{SN_s^{MAX}}$ has already been received (line 13). Going back to the basic algorithm (Fig. 6), if $p.\mathtt{SN} \leq \mathtt{SN_s^{MAX}}$ then v should decide whether $p.\mathtt{SN}$ is one of the k most recent sequence numbers. If not (lines 5-7) the packet is dropped because it is not possible to decide whether it is a duplicate or not. Otherwise, v calculates the index of the bit that corresponds to $p.\mathtt{SN}$ (lines 9-12). If that bit is set to 1 then p is dropped because it is a duplicate otherwise the bit is set to 1 and p is relayed if v is an elected forwarder (lines 13-18).

To validate the efficacy of MC/U we developed a version of the PDP scheme that utilizes it instead of C/U. Then, we repeated the same experiments described in Section 3.2, testing different values of RAD under varying offered load. Fig. 8 illustrates our main results. In contrast to PDP C/U (Fig. 4a),

6

RAD has a negligible impact on the delivery performance of PDP MC/U (Fig. 8a). At the same time, the ability of MC/U to prune transmissions is not damaged. Both MC/U and C/U achieve roughly the same number of transmissions (compare Fig. 8b and 4b when the performance of C/U does not collapse, i.e. when no RAD is used). Overall, the results prove that MC/U successfully tackles the packet reordering problem.

5. Network Coding Broadcast with Coded Redundancy

In this section, we introduce the *Network cOding Broadcast with Coded Redundancy* (NOB-CR) algorithm. NOB-CR, similar to other schemes, takes the approach to implement XOR coding on top of a CDS-based broadcast algorithm. However, in order to maximize the performance of network coding, NOB-CR employs the MC/U termination criterion introduced in section 4. As the default CDS algorithm we choose PDP although any algorithm of this category could be used. Regarding the coding process, similar to XOR coding approaches, NOB-CR utilizes network coding on a hop-by-hop basis. Each intermediate node uses bitwise XOR operations to combine native packets into encoded ones under the requirement that all neighboring nodes can decode them. Nonetheless, NOB-CR uses only one of the two specialized data structures required for coding (see Section 2.2), i.e. the packet pool. This is because its lightweight coding detection mechanism renders obsolete the use of the other one, i.e. the neighbor reception table. We discuss this issue in detail in Section 5.2. Besides the aforementioned differences, NOB-CR deviates from other approaches by introducing a series of mechanisms that significantly improve the broadcasting performance and alleviate the related costs. In particular, these mechanisms are:

- A lightweight coding detection method that operates without the need of maintaining a neighbor reception table.

- A novel method for the computation of forwarders that uses information provided by the RAD mechanism to increase the overall pruning efficiency.

- A cost-free method to inject packet redundancy in the network in order to reduce the end-to-end delay.

In the following, we delineate NOB-CR's basic operation as well as the aforementioned mechanisms.

5.1. Basic operation

Fig. 9 describes the basic operation of NOB-CR by illustrating the life cycle of a packet at a node that implements the protocol. An incoming packet is first examined for deciding if it is a native or an encoded one. In the latter case, the packet is decoded to produce the native packets that it consists of. Then, for each native packet the receiving node examines: a) whether the packet meets the termination criterion conditions, and b) the set of forwarders that is piggybacked on the packet to determine if it is selected as a forwarder for the packet. If at least one of these tests is negative the packet is dropped and the process terminates. Otherwise, the coding opportunity detection

Figure 9: Flow diagram of a packet's life cycle in NOB-CR.

process initiates. If a coding opportunity is detected, the set of forwarders is determined (please refer to Section 2.2 for details) for each native packet involved and then the encoded packet is created and immediately transmitted. In the absence of a coding opportunity, the received packet is temporarily buffered in the output queue for a randomly chosen time interval according to the RAD mechanism. This allows the packet to participate in subsequent coding inquiries. When the buffering interval expires the packet is examined for coding one last time before being transmitted as native.

5.2. Lightweight detection of coding opportunities

Finding coding opportunities at an intermediate node strongly depends on the packets that each neighbor has already received. To acquire such information, each node snoops all communication in the wireless medium [4, 29] and stores it in the neighbor reception table. However, the maintenance of this table comes at a significant cost. Keeping track of every packet received by each neighbor requires a considerable amount of storage. Likewise, updating the neighbor reception table on a packet arrival basis is a task that requires significant processing power. In order to avoid these costs, we introduce a new approach for finding coding opportunities. Our method operates without the need for a neighbor reception table. In fact, it uses neighborhood information that is available through the underlying broadcasting mechanism. To explain the method, let us use the example in Fig. 10a. In this, node v receives packets

Figure 10: Searching for coding opportunities when v has received (a) one duplicate, or (b) more duplicates per packet.

p_1 and p_2 from u and w respectively and checks for a coding opportunity. Recall that a coding opportunity exists only when all of v's neighbors can decode the prospective encoded packet $p_1 \oplus p_2$. In other words, all neighbors must know either p_1 or p_2, or both. Observe that the set of neighbors that can decode the packet is $\mathcal{N}(u) \cup \mathcal{N}(w)$. As a result, it suffices for v to confirm that the set $\mathcal{R} = \mathcal{N}(v) - \mathcal{N}(u) - \mathcal{N}(w)$ is an empty set to decide that $p_1 \oplus p_2$ is possible.

To increase the probability of producing an encoded packet our approach takes advantage of packet duplicates. Recall that a native packet waits in node v for a random time in order to find a coding opportunity. During this time v receives multiple copies of the native packet. Our observation is that each of these copies reaches a different part of v's neighborhood. As a result, it is possible to minimize $\mathcal{R}$ and thus increase the probability of finding a coding opportunity. Fig. 10b illustrates the advantages of considering packet duplicates. In the example, node v receives a duplicate of p_1 from node z while initially v received p_1 from node u. Taking into account the neighbors that indirectly received p_1 through node z, node v searches for coding opportunities by estimating the set $\mathcal{R} = \mathcal{N}(v) - \mathcal{N}(w) - (\mathcal{N}(u) \cup \mathcal{N}(z))$, which is now an empty set and therefore coding is possible. In general, when multiple duplicates of both p_1 and p_2 exist, v can detect coding opportunities using the set

$$\mathcal{R} = \mathcal{N}(v) - \mathcal{Z}_1 - \mathcal{Z}_2 \qquad (1)$$

where

$$\mathcal{Z}_m = \bigcup_{i \in \mathcal{H}_m} \mathcal{N}(i) \qquad (2)$$

and $\mathcal{H}_m$ is the set containing all of v's neighbors that forwarded a copy of packet p_m.

An important feature of a coding process is to be able to find coding opportunities that involve more than two native packets, i.e. increase what is known as the *coding depth*. This feature is critical because it allows for further reduction of transmissions, thus improving energy efficiency. To illustrate that it is possible to use the proposed method to find coding opportunities involving multiple packets let us extend the example in Fig. 10b. Assume now that another native packet p_3 is available at node v and that we wish to check whether we can include it in the original coding $p_1 \oplus p_2$, i.e. create the encoded packet $p_1 \oplus p_2 \oplus p_3$. Recall that the prerequisite is that every node in $\mathcal{N}(v)$ should know at least two of the thee packets. Observe that nodes in the set $\mathcal{S} = \mathcal{N}(w) \cap (\mathcal{N}(u) \cup \mathcal{N}(z)) = \mathcal{Z}_1 \cap \mathcal{Z}_2$ have received both p_1 and p_2 therefore they fulfill the prerequisite. On the other hand, nodes in $\mathcal{N}(v) - \mathcal{S}$ (gray area in Fig. 10b) do not know both packets but have received either p_1 or p_2 (otherwise the coding of p_1 and p_2 could not be possible). Consequently, these nodes should known about p_3 in order for the triple coding to be possible. In other words, the set $\mathcal{R} = \mathcal{N}(v) - \mathcal{S} - \mathcal{Z}_3$ should be an empty set. The process can be repeated recursively to include more native packets in the encoding. In general, in order to include a native packet p_n in an encoding that already contains packets $p_1, p_2, \ldots, p_m$ node v should check whether

$$\mathcal{R} = \mathcal{N}(v) - \mathcal{S} - \mathcal{Z}_n = \mathcal{N}(v) - \bigcap_{j=1}^{m} \mathcal{Z}_j - \mathcal{Z}_n \qquad (3)$$

DetectCodingOpportunities(packet p, output Queue Q)

1: $\mathcal{Z}_p = GetReceiversOf(p)$
2: $\mathcal{S} = \mathcal{Z}_p$
3: $C = \mathcal{N}(v) - \mathcal{S}$
4: $e = p$
5: **for** each native packet $\mathbf{q} \in Q$ **do**
6:     $\mathcal{Z}_q = GetReceiversOf(q)$
7:     $\mathcal{R} = C - \mathcal{Z}_q$
8:     **if** ($\mathcal{R} = \emptyset$) **then**
9:         $\mathcal{S} = \mathcal{S} \cap \mathcal{Z}_q$
10:        $C = \mathcal{N}(v) - \mathcal{S}$
11:        $e = e \oplus q$
12:     **end if**
13: **end for**
14: **return** e

Figure 11: Pseudocode for detecting coding opportunities at node v.

is an empty set.

Fig. 11 presents the pseudocode of NOB-CR's coding procedure. As we mentioned previously, our method renders the use of a neighbor reception table obsolete which significantly alleviates the related costs. Instead, we mostly rely on information already available through the underlying broadcast algorithm, i.e. neighborhood information. The only additional information that the method requires for a packet p is the set of previous hops $\mathcal{H}_p$, i.e. the nodes that forwarded a copy of p to v. This information can be used to estimate the set $\mathcal{Z}_p$ (procedure $GetReceiversOf(p)$ in the pseudocode) which consists of the nodes that have received p. This can be done using neighborhood information, i.e. $\mathcal{Z}_p = \bigcup_{\forall i \in \mathcal{H}_p} \mathcal{N}(i)$. Bear in mind that neighborhood information is updated on a periodic basis. Furthermore, each packet remains to the output queue for a limited period of time which is smaller than typical values for a neighborhood update interval. Therefore estimating the receivers of p by using $\mathcal{Z}_p$ is as accurate as the neighborhood information. However, note that due to periodic updating and mobility, $\mathcal{Z}_p$ is an approximation of the nodes that actually received p. In Section 6 we evaluate the impact of mobility on our coding approach and show that it is negligible even in networks of high node mobility. In any case, it is also possible to use the traditional approach for finding coding opportunities in NOB-CR. Clearly, the advantage of our method is the limited cost for storing and updating $\mathcal{H}_p$. Note that, each packet is queued for a limited period of time. Therefore the size of $\mathcal{H}_p$ is expected to be rather limited thus contributing in the reduction of the related costs. Finally, it is possible to represent all required information using bitmaps in order to minimize storage and at the same time benefit from fast set operations.

5.3. Exploiting RAD to enhance the pruning process

The pruning efficiency of the underlying CDS-based algorithm is equally important to the coding process for minimizing transmissions. In general, algorithms that use the source-based CDS approach are considered to be the most efficient. In fact, NOB-CR builds on top of such an algorithm, i.e. PDP. Recall that in PDP a node v elects forwarders from $C(v) = \mathcal{N}(v) - \mathcal{N}(u)$ in order to deliver a packet p to all nodes in $\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v) - \mathcal{N}(u) - \mathcal{N}(\mathcal{N}(u) \cap \mathcal{N}(v))$, where u is the previous hop node that forwarded p to v. We take advantage

of information already provided by the RAD mechanism to further enhance the pruning efficiency of PDP. More specifically, we make the observation that, when electing the forwarders, it is possible to take into account not only the previous hop node of p but all other nodes that relayed a duplicate of p while it was buffered due to the RAD mechanism. As a result, the set of nodes to be covered can be further reduced to:

$$\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v) - \bigcup_{i \in \mathcal{H}_p} \mathcal{N}(i) - \bigcup_{i \in \mathcal{H}_p} \mathcal{N}(\mathcal{N}(i) \cap \mathcal{N}(v)) \quad (4)$$

where the set $\mathcal{H}_p$ is the set consisting of all previous hop nodes of p. The same idea can be applied to any source-based CDS algorithm that uses the same problem modeling such as DP, TDP and their derivatives. It is clear that reducing $\mathcal{U}(v)$ increases the probability of selecting fewer forwarding nodes from the candidate set $C(v)$. Also note that the proposed method comes at no additional cost since the set $\mathcal{H}_p$ is used for detecting coding opportunities.

5.4. Reducing delay through coded redundancy

The performance of XOR-based broadcast schemes heavily depends on the random assessment delay (RAD) applied before relaying a packet. In particular, high RAD values increase the coding opportunities and maximize the coding gain. This behavior was confirmed by the experimental results in Section 3.1 (Fig. 1 and Fig. 2) where increasing the RAD value from 200 to 400 *ms* substantially enhances CodeB's performance in terms of delivery ratio and energy efficiency. However, imposing a RAD to a packet in every node has a major impact on the end-to-end delay. Although RAD is usually short it results to an aggregation of a considerable end-to-end delay. To highlight this effect, we replayed the first experiment in Section 3.1 and recorded the end-to-end-delay for different values of RAD (Fig. 12). To avoid any interference caused by the performance degradation of the M/U criterion under high load we used the PDP algorithm with the MC/U criterion which can sustain performance in such conditions (Fig. 8, Section 4).

The results confirm our observation and expose the paramount importance of reducing the end-to-end delay when employing RAD. One way towards this direction is to increase the packet redundancy across the network. The rationale is that
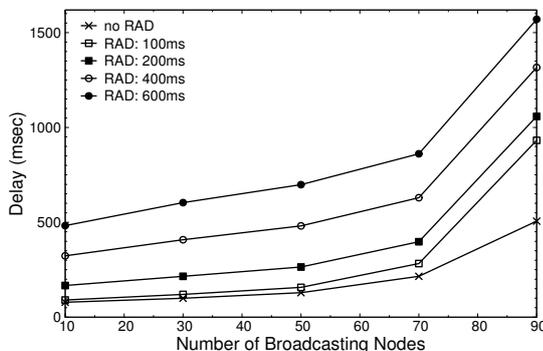

Figure 12: End-to-end delay performance of PDP MC/U under varying traffic load using different RAD intervals

using more duplicates per packet increases the probability of delivering a copy of the packet through a faster path. However, the practice of increasing redundancy should be exercised with caution because it usually results in extra transmissions. This impacts the energy efficiency of the broadcast process as well as its delivery efficiency through the increase of collisions.

We propose a cost-free method for introducing packet redundancy across the network. This method, called *Coded Redundancy (CR)*, targets at increasing the duplicates of a packet and it is cost-free in the sense that it does not produce new transmissions. To accomplish that, CR introduces a new packet type called *gratis*. Gratis packets are non-coded packets for which the receiving node has not been selected for relaying them. Instead of dropping them, our method examines if these packets could be forwarded as part of already encoded packets, i.e. without cost. Packets already delivered to all neighbors of a node v do not qualify for marked as gratis because no delay improvement is feasible. Summarizing, node v can mark gratis packets using the following criterion:

Definition 1 (Gratis Marking Criterion). *A native packet p is marked as gratis by a node v if there is at least one neighbor of v that has not received p and v is not a forwarder of p.*

Note that it is possible for v to estimate whether a packet has not been received by all of its neighbors by utilizing a similar methodology as the one used for detecting coding opportunities (Section 5.2).

Many aspects of packet handling are the same for gratis and native packets. More specifically, in order to avoid loops, gratis packets are considered for forwarding only if they conform to the termination criterion. Accepted gratis packets are also temporarily buffered using the RAD technique. Then, if a coding opportunity is detected one or more gratis packets are relayed as part of an already encoded packet. Nonetheless, there are also differences in handling gratis and native packets. The first is that gratis packets are dropped if their buffering time expires without finding a coding opportunity. This is because they are meant to be forwarded only as part of an encoded packet. For the same reason gratis packets do not participate in the forwarding process, i.e. a node does not determine a set of forwarders for a gratis packet. Instead, as soon as a packet is marked as gratis it is always treated as gratis in subsequent hops. This approach guarantees that gratis packets are forwarded without any additional cost. Finally, each node applies the following rule upon reception of a gratis packet:

Definition 2 (Gratis Receiving Rule). *The arrival of a gratis packet never triggers any modification to the termination criterion structures.*

This rule is directly associated with the termination criterion and indirectly affects the other packet duplicates that coexist in the network. If not applied, it could lead to situations where packets prematurely terminate their propagation in the network destroying the protocol's delivery efficiency. We further investigate the importance of this rule through an example. In Fig. 13, we monitor a packet p_1 that is propagated through a part of a
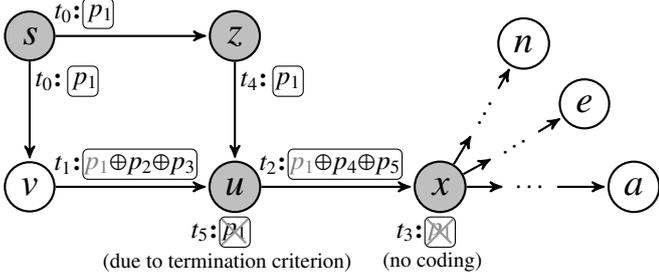
9

Figure 13: Example of failing to deliver packet p_1 to all network nodes when the "Gratis Receiving Rule" is not implemented.

network. Nodes in gray are selected as forwarders for p_1, while all other nodes act as passive receivers. Initially, at time t_0, nodes v and z receive a duplicate of p_1 from the source node s. Node v handles p_1 as gratis since it is not elected as a forwarder while for the opposite reason node z treats p_1 as native. At some point in time (t_1), node v detects a coding opportunity between the gratis p_1 and the already encoded packet $p_2 \oplus p_3$. As a result, it transmits the encoded packet $p_1 \oplus p_2 \oplus p_3$. Node u receives the encoded packet and successfully decodes it (assuming p_2 and p_3 are already known). Node u also handles the copy of p_1 as gratis. At time t_2 a coding opportunity for gratis packet p_1 at node u results in the transmission of a new encoded packet, i.e. $p_1 \oplus p_4 \oplus p_5$, where p_4 and p_5 are ordinary native packets previously received at node u. Node x receives the encoded packet and decodes it (assuming p_4 and p_5 are already known). Likewise, x handles p_1 as gratis searching for a proper coding opportunity to relay it. Assuming that p_1's buffering time expires with no coding opportunities, x drops p_1 (time t_3), terminating its dissemination to the rest of the network. At this point, the only way to deliver p_1 to nodes n, e and a is through the duplicate of p_1 that node z holds. At time t_4, the duplicate of p_1 is transmitted as a native packet because z is an elected forwarder for p_1. Node u, which is also an elected forwarder, successfully receives p_1 and has the opportunity to further relay it. However, its decision depends on the previous reception of the gratis copy of p_1 at time t_1. In particular, u employs the MC/U termination criterion which allows a node to relay only packets seen for the first time. In case that u has recorded the former arrival of p_1 at t_1 no forwarding is allowed due to the termination criterion. Consequently, the newly fetched duplicate of p_1 is dropped (time t_5). On the other hand, implementing the "Gratis Receiving Rule" resolves the situation. According to the rule, the former arrival of p_1 as a gratis packet is never registered by node u. As a result, the native copy of p_1 is forwarded to node x (time t_5). From that point, node x successfully propagates p_1 across all parts of the network.

The example in Fig. 13 also illustrates the benefits of using the coding redundancy technique. Packet p_1 reaches nodes u and x much faster when coding redundancy is utilized. More specifically, nodes u and x receive p_1 at time instances t_1 and t_2, respectively. On the contrary, without coded redundancy, p_1 reaches nodes u and x at t_4 and t_5, respectively.

The key functionality of the coding operation is the detection of coding opportunities. Extending this functionality to support gratis packets is not straightforward. This is because there are cases where gratis packets could destroy coding opportunities involving native ones, thus impairing the protocol's energy efficiency. Let us examine this problem through an example. Suppose that an intermediate node v is elected as forwarder for two native packets, i.e. p_1 and p_2, and that these packets can be mixed together (forming the encoded packet $p_1 \oplus p_2$) and forwarded with a single transmission. Things get complicated when a gratis packet p_3 is also present at v. Assuming that p_3 can be combined only with p_1 and that v chooses to combine p_1 with p_3 instead of p_2 then two transmissions are required, i.e. one for $p_1 \oplus p_3$ and one for p_2. To evade this problem, we introduce the following rule that every node applies when searching for coding opportunities.

Definition 3 (Gratis Coding Rule). *Gratis packets participate in the coding process with lower priority than native packets.*

Furthermore, in case that no coding opportunity is found among native packets then coding detection terminates without examining gratis packets. This policy in conjunction with the above rule prevents a gratis packet from directly being combined with existing native ones, leaving them available for possible future encodings. The only exception of combining a gratis packet with a native one is when the buffering time of the latter expires. In that case there is no possibility for the native packet to participate in any future coding opportunity. As a result, encoding the native packet with the gratis one does not destroy future coding opportunities.

6. Evaluation

To evaluate the performance of NOB-CR, we compare it with two algorithms. The first one is CodeB [4] which is the most representative of XOR coding-based broadcast algorithms. The second algorithm is PDP [33] which, despite the fact that does not use any type of coding, is well-known for its energy efficiency. We use two variants of PDP, namely PDP M/U and PDP C/U, in order to examine how the termination criterion affects the overall performance.
Set up and methodology: All investigated algorithms are implemented in the ns2 simulator [43], using the CMU extension. We

Table 1: Simulation parameters

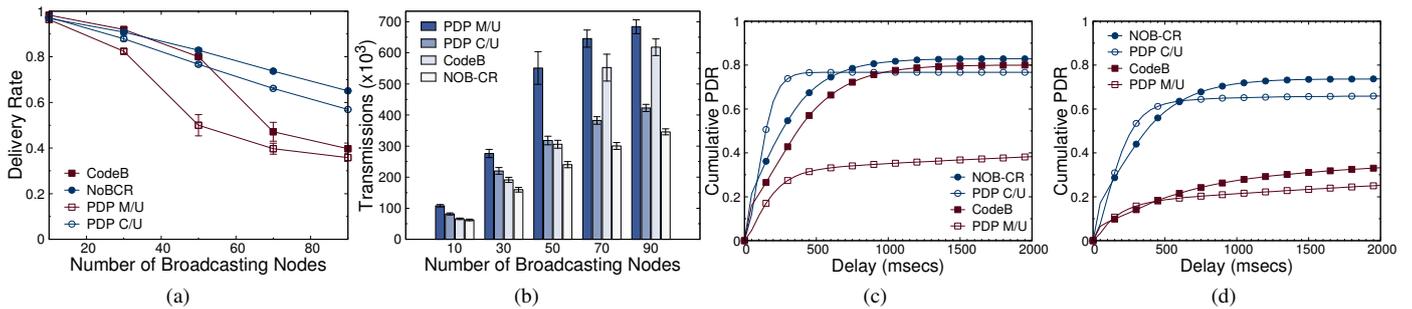|	
Simulation Time	300 sec
Number of Trials	20
Confidence Level	95%
Transmission Range (R)	250 m
Bandwidth	2 Mb/sec
Number of Nodes (N)	60 - 250
Avg. Neigh. Size	15, 30
Node Speed	0 - 20 m/sec
Broadcast Sessions (S)	10 - 90
Broadcast Rate (λ)	0.1 - 8 pkts/sec
Packet Size	256 Bytes
Hello Interval (t_H)	1 sec
Random Assessment Delay (t_{RAD})	100 - 600 msec

10

Figure 14: Performance for different levels of offered load in the "sparse" topology (N=100, max speed:1 m/sec, t_{RAD}=400msec): (a) Delivery rate vs broadcasting sources (b) Avg. number of transmissions vs broadcasting sources (c) Cumulative PDR vs end-to-end delay (S=50 sources) (d) Cumulative PDR vs end-to-end delay (S=70 sources).

present the average values over 20 independent simulation runs, each with a duration of 300 seconds. The confidence level, for all reported confidence intervals, is 95%.

Network model: The default number of nodes is 100, the propagation model is the TwoRay ground with a transmission range of 250m and the nominal bit rate is 2Mbps. The nodes move in a square area according to the Random Waypoint (RW) model [44]. To avoid transient artifacts in nodes' movement, we use the perfect simulation algorithm [46]. We examine two network topologies; "dense" and "sparse". Similar to [4], in the "dense" topology, the average neighborhood size is 30 while in the "sparse" topology it is 15. Note that we could not use a lower density in the "sparse" scenario since in such a case frequent partitions occur. Simulations confirmed that in the "sparse" scenario there exist many nodes (those moving near the boundaries) that experience very low connectivity. All algorithms collect neighborhood information by periodically exchanging hello messages with an interval (t_H) of 1 second.

Network traffic: Traffic is generated by broadcast sessions, each stemming from a different source node and starting at a random time. Although we use a variable number of sources, each one producing packets at a constant rate of λ = 1pkt/sec, the default value is 50. The size of each message is set to 256 Bytes.

Coding parameters: All coding schemes under evaluation use the RAD technique to maximize the probability of coding opportunities. According to RAD, each node delays every packet it receives for a random delay in $[0, t_{RAD}]$. The default t_{RAD} value in our simulations is 400 msec. Due to storage limitations, all coding based algorithms buffer incoming packets for a limited time interval (B_T) in order to enable decoding. This time interval is highly correlated with the time period for which information inside the neighbor reception table is available (R_T). We set both time intervals to 5 sec for the CodeB algorithm in order to increase the benefits of network coding and be realistic at the same time. On the contrary, we set only the B_T interval for NOB-CR algorithm since it operates without a neighbor reception table. After extensive experimentation, we found that a B_T value equal to 2 sec is sufficient for NOB-CR's encoding/decoding operation. The small B_T value utilized by NOB-CR is preferable because it allows for efficient management of the limited storage space in the network nodes.

Fig. 14 illustrates the performance of all investigated algorithms in the sparse topology and under different levels of offered load (variable number of broadcasting sources). As discussed in section 3.1, our experimental results reveal the ineffectiveness of the M/U criterion that induces the performance breakdown of the PDP M/U and CodeB schemes. More specifically, as the load increases the algorithms that utilize the M/U criterion lose their pruning efficiency producing a large number of transmissions (Fig. 14b). PDP M/U fails to prune transmissions when the number of source nodes exceeds 30. On the other hand, network coding enables CodeB to maintain its pruning efficiency when traffic is produced by up to 50 sources (half of network nodes). However, as the congestion level increases, the excessive number of forwarding decisions taken by both algorithms induce transmission failures due to packet collisions. As a result, their delivery performance deteriorates (Fig. 14a). NOB-CR outperforms all algorithms both in terms of delivery efficiency and number of transmissions. Even in the extreme case of 90 broadcasting sources, which is close to the all-to-all communication paradigm, NOB-CR delivers ~66% of the traffic while the M/U based schemes reach less than 40% of the network nodes (Fig. 14a). This justifies our approach to combine XOR network coding with an termination criterion other than M/U. At the same time, NOB-CR is exceptionally energy efficient. Table 2 presents NOB-CR's energy gains that derive from reducing the total number of transmissions. Against PDP variants, NOB-CR reduces the total number of transmissions by 21% in the worst case. Compared to CodeB, the energy gains become noticeable when the broadcasting sources are more than 30 (18% to 45%).

Fig. 14c and 14d depict the cumulative packet delivery ratio (PDR) versus the end-to-end delay, i.e. the fraction of packets

Table 2: Reduction (%) of transmissions for NOB-CR compared to other algorithms

# sources	PDP M/U	PDP C/U	CodeB
10	44.7%	27.0%	7.7%
30	44.1%	29.8%	17.9%
50	57.8%	26.9%	22.7%
70	55.1%	24.2%	45.5%
90	51.3%	21.1%	45.3%

delivered within a delay limit, when the broadcasting sources are 50 and 70, respectively. We choose this presentation style in order to capture both the delivery efficiency and the timeliness of each algorithm. Again, the results provide a confirmation of the ineffectiveness of the M/U criterion. Even when there are only 50 sources in the network (Fig. 14c) PDP M/U collapses while XOR coding allows CodeB to achieve a competitive performance. However, when the offered load increases (Fig. 14d), both schemes that use the M/U criterion deliver less packets with higher delay due to the increased number of transmission failures. NOB-CR outperforms both schemes not only in delivering more packets, but in delivering them faster. The reason is twofold; the utilization of the MC/U criterion and the coded redundancy mechanism. Interestingly, NOB-CR's delay profile is comparable to that of PDP C/U (especially in the case of 70 sources) despite the fact that simple PDP schemes operate without the RAD mechanism that significantly increases the end-to-end delay. Moreover, we found that under high load (> 70 sources) NOB-CR is at least as fast as PDP C/U. This is due to NOB-CR's pruning process that efficiently decreases transmission failures allowing for timely packet delivery.

We also experimented on increasing the packet generation rate λ while keeping the number of sources constant, e.g. S=10 sources. In this way it is possible to change the offered load but limit the coding capability of algorithms. This is because coding opportunities heavily depend on the number of packet flows, i.e. the number of sources. The obtained results are qualitatively similar to the previous experiment therefore we do not include the corresponding performance plots. In summary, for a low λ, the energy gains for all coding enabled schemes are limited. This is because the number of coding opportunities is rather small since fewer packets coincide in the network. As the offered load increases, the benefits of network coding become more evident. However, after the breaking point of λ=5 packets per second the delivery performance of CodeB deteriorates as a result of the increased levels of congestion. On the other hand, NOB-CR exhibits a remarkable resilience to congestion achieving the best performance in terms of delivery ratio and energy efficiency.

Next, we used a variable number of sources to test all schemes under different levels of offered load in the dense topology (Fig. 15). As expected, when the offered load is low, the delivery efficiency of all algorithms improves compared to the sparse topology (Fig. 15a). This is because the higher number of neighbors results in increased packet redundancy, making delivery more probable. Furthermore, the diameter of a denser network is smaller. Consequently, as depicted in Fig. 15a, all schemes deliver packets faster than in the sparse case (Fig. 14c). However, despite its positive effects, there is also a downside of the increased neighborhood size; under high load the probability of transmission failures due to collisions is higher. This is because more packet duplicates are created, resulting in congestion. As a result, schemes that do not efficiently prune transmissions collapse (Fig. 15b) as the offered load increases. Interestingly, the performance degradation is more acute and develops more quickly (lower traffic levels) than in the case of sparse topology because congestion is more severe. Fig. 15c illustrates
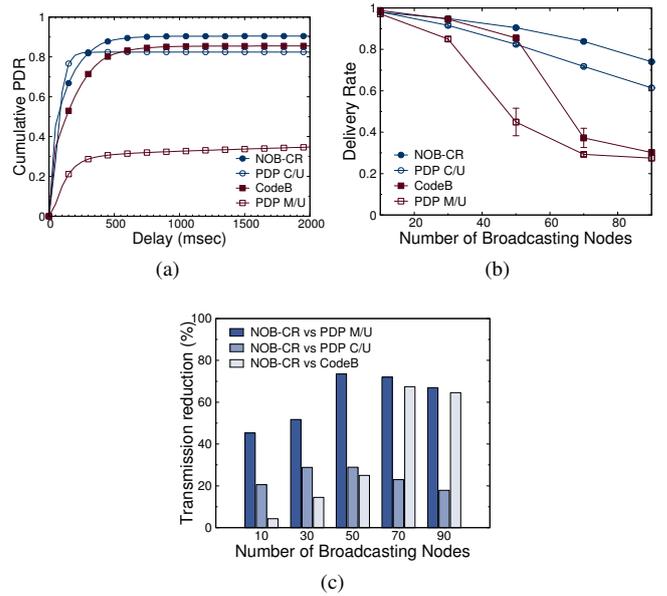

Figure 15: Performance for different levels of offered load in the "dense" topology (N=100, max speed:1 m/sec, t_{RAD}=400msec): (a) Cumulative PDR vs end-to-end delay (S=50 sources) (b) Delivery rate vs sources (c) Transmission reduction vs sources

the reduction of the total number of transmissions achieved by NOB-CR compared to all other schemes. As anticipated, the higher gains are witnessed when the offered load is high where NOB-CR prunes 60% more transmissions than CodeB. At the same time, it delivers over 40% more packets (Fig. 15b).

In the following we focus on the more challenging scenario of sparse networks. The next set of experiments assesses the performance of NOB-CR and CodeB when scaling the network up. Towards this direction, we conducted simulations with an increasing number of nodes. At the same time, we also expand the network area in which the nodes move in order to keep the average neighborhood size fixed. Fig. 16 illustrates the cumulative packet delivery ratio versus the end-to-end delay for various network sizes. NOB-CR and CodeB present a similar behavior when the network size is small, i.e. 60 nodes. As the number of nodes increases, the performance of CodeB quickly deteriorates and finally collapses when the network size exceeds 140 nodes (Fig. 16b). On the other hand, NOB-CR exhibits a remarkable durability and its performance, in terms of both de-
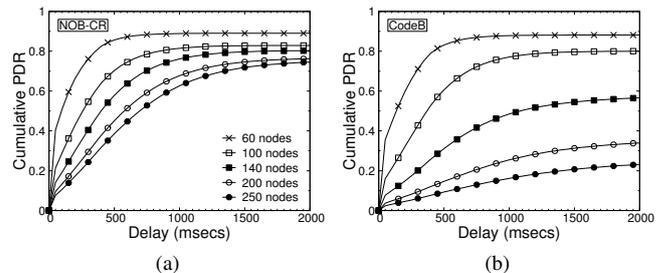

Figure 16: Performance when the network size scales up ("sparse" topology, S=50 sources, max speed:1 m/sec, t_{RAD}=400msec): Cumulative PDR vs end-to-end delay for a) NOB-CR, and (b) CodeB.
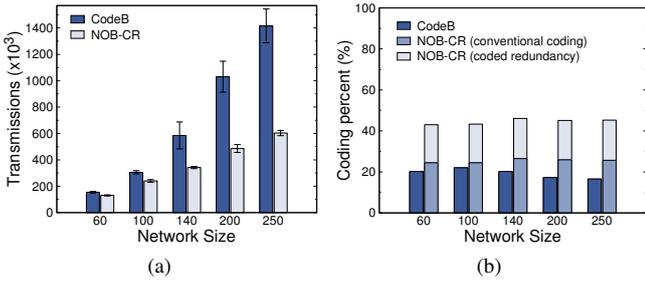
Figure 17: Performance when the network size scales up ("sparse" topology, $S = 50$ sources, max speed:1 m/sec, $t_{RAD} = 400$msec): (a) Avg. number of transmissions, and (b) Fraction of encoded packets vs network size.

Figure 18: Performance under different levels of mobility ("sparse" topology, N=100, S=50 sources, t_{RAD}=400msec): Cumulative PDR vs end-to-end delay for Node speed: (a) $2 - 10$ m/sec (b) $10 - 20$ m/sec.

Table 3: NOB-CR's energy gains over CodeB under different mobility levels

Mobility	$0 - 1$ m/sec	$2 - 10$ m/sec	$10 - 20$ m/sec
Gain	21.45%	23.94%	27.83%

livery ratio and end-to-end delay, degrades much more slower and smoother (Fig. 16a). The witnessed performance degradation for both schemes is reasonable since in our experiment we fix the network density. As a result, the diameter of the network increases with the number of nodes and therefore it is more difficult to reach some destinations. Notwithstanding, NOB-CR is very efficient in reducing transmissions (Fig. 17a) and therefore alleviates congestion. Thus, failures due to collisions are minimized and so is the impact of the increasing network diameter. For example, in the case of 60 nodes NOB-CR produces ~16% less transmission than CodeB, while in case of 140 nodes transmissions are reduced by ~42%. This increasing difference in the pruning efficiency of the two schemes not only justifies the higher deliver ratio of NOB-CR but is also in accordance with the increasing difference in the delivery efficiency of the two schemes. Overall, NOB-CR loses less than 15% of its delivery efficiency in the most demanding scenario of 250 nodes. On the other hand, in the same scenario CodeB loses more than 60%.

Going back to Fig. 17a, it is worth pointing out that the pruning efficiency of NOB-CR is increasingly better compared to that of CodeB. As expected, the total number of transmissions increases because more forwarders are required for a bigger network with a larger diameter. However, NOB-CR manages to suppress this increase and therefore broaden its advantage over CodeB. The reason for this result is not only the better pruning operation of the MC/U termination criterion but also its more efficient coding operation. To illustrate this, we present in Fig. 17b the number of transmitted encoded packets as a percentage of the total (encoded and native) number of transmitted ones. Furthermore, in the case of NOB-CR we present two separate classes on encoded packets. The one consists of encoded packets containing at least one gratis packet (Coded Redundancy) while the other refers to typical encodings involving only native packets (conventional coding). Recall that besides the packets encoded with the conventional mechanism, those produced with the Coded Redundancy method may also reduce the number of transmissions. This is because the latter packets may also contain two or more native packets. Clearly, NOB-CR not only consistently performs about twice as many encodings as CodeB does, but its coding operation is also stable. In contrast, the percentage of coded packets for CodeB decreases in bigger networks which implies that the coding operation is hampered by the underlying broadcast mechanism.
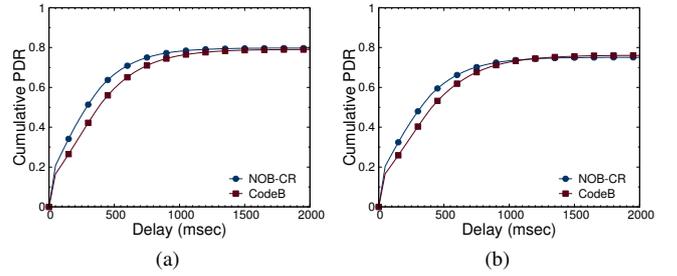
Another important finding is that NOB-CR also performs a significant number of encodings involving gratis packets. Although this type of encodings is important for reducing end-to-end delay (as discussed in Section 5.4), it is also beneficial for enhancing delivery efficiency because gratis packets increase packet redundancy across the network without any additional cost. Therefore, the Coded Redundancy mechanism also contributes to NOB-CR's superior delivery efficiency witnessed in Fig. 16.

In the last experiment we assess the delivery efficiency under different levels of mobility (Fig. 18). Clearly, increased mobility impacts the performance of both NOB-CR and CodeB. The reason is that both schemes rely on the PDP scheme. The latter uses neighborhood information for electing the optimal forwarders. This information becomes outdated more quickly when mobility increases. Note that, as discussed in Section 5.2, NOB-CR uses neighborhood information also for detecting coding opportunities while CodeB uses a neighbor reception table. Nonetheless, NOB-CR manages to outperform CodeB regardless of the mobility level and even in scenarios of very high mobility. More specifically, it exhibits faster packet delivery (Fig. 18) using more than ~20% less transmissions compared to CodeB (Table 3). Finally, it is worth pointing out that it is possible for NOB-CR to deactivate the lightweight coding process discussed in Section 5.2 and use the typical neighbor reception table based approach.

7. Related work

Several studies have explored the use of network coding for broadcasting in wireless ad hoc networks. The proposed algorithms can be classified into: i) energy efficient [4–13], and ii) delivery guarantee [14–25] approaches. The first category aims at striking the best possible trade-off between energy expenditure (usually expressed by the number of transmissions) and delivery efficiency. On the other hand, the second category targets at 100% packet delivery and treats the minimization of the related costs as a second priority task.

As we mentioned, in this work we focus on energy efficient broadcasting. The proposed approaches in this field can be further classified, based on the coding method, into: i) XOR-based, and ii) RLNC-based approaches. The first subclass of algorithms [4–9] follows the concept of "coding opportunity" [29] to combine multiple packets into an encoded one using bitwise XOR operations. The coding process is performed on a hop-by-hop basis, i.e. decoding takes place at every hop. The prominent algorithm of this subclass, CodeB [4], combines CDS-based broadcasting with XOR network coding. It also provides information exchange mechanisms that make possible the implementation in mobile environments. CodeB builds on top of the non coding PDP [33] scheme, which is actually a CDS-based broadcast algorithm. However, it can be directly applied to other CDS-based approaches. Wang et al. explore the benefits of employing XOR network coding on various underlying CDS-based broadcast schemes [5, 6]. Moreover, the use of XOR coding over PDP and MPR [31], two typical CDS based algorithms, in tactical networks has been studied in [7]. In [8] the authors study the problem of broadcasting with deadlines in static ad hoc networks and propose alternative buffering schemes for the RAD mechanism. Finally, Yang and Wu [9] explore the benefits of XOR coding in energy efficient broadcasting when combined with directional antennas.

The second subclass of the energy efficient category consists of RLNC-based algorithms [10–13]. These algorithms build on the concepts of practical random linear network coding [47]. Encoded packets are created using random linear combinations based on the theory of finite fields [48, 49]. Then, they are forwarded either probabilistically [10–12] or deterministically using a CDS-based algorithm [13]. Coding is allowed only between packets in the same group, called "packet generation", and is performed in an end-to-end basis, i.e., encoded packets are decoded at the prospective receivers and only when a sufficient amount of encoded packet is gathered. These algorithms require special mechanisms for allowing nodes to distributively agree on the grouping of packets into generations.

In the second major category of network coding enabled broadcast algorithms the focus is on guaranteeing 100% reliability. Here, there are also two subclasses that can be identified. The first consists mainly of XOR-based algorithms that adopt a rateless approach [14–17]. Algorithms that utilize rateless coding keep producing encoded packets until all receivers are capable of decoding the initial packets. As a result, these schemes require feedback information. In any case, implementing a feedback mechanism is not straightforward in mobile networks. Therefore, those algorithms are limited to static networks. The same holds for the second subclass of algorithms that also follow a rateless approach but utilize random linear network coding [18–25]. Furthermore, most algorithms in this subclass use only intra-source coding, i.e. encoding packets only from the same source which significantly limits the coding gains in a multi-source scenario [12].

8. Conclusion

XOR-based coding has been successfully used to enhance the energy efficiency of broadcasting in mobile ad hoc networks. We demonstrated, through extensive experimentation, that using the M/U termination criterion in the baseline broadcast algorithm severely impairs performance in several cases. Unfortunately, we found that alternative termination criteria that are proposed in the literature are not capable of efficiently supporting the coding process. As a result, we introduced a novel termination criterion that is fully compatible with XOR-based network coding. Furthermore, we revised some of the coding internals to enhance performance and at the same time reduce complexity. More specifically, we proposed a lightweight method for detecting coding opportunities that operates without a reception table. We introduced the concept of "Coded Redundancy" that reduces the end-to-end delay by increasing the packet redundancy across the network at no additional cost. Finally, we improved the forwarder election process of the proposed algorithm by exploiting information that was originally used only for coding purposes. The efficiency of NOB-CR, the algorithm that incarnates all the aforementioned modifications, was demonstrated through extensive simulations.

References

References

[1] M. Abolhasan, T. Wysocki, E. Dutkiewicz, A review of routing protocols for mobile ad hoc networks, Elsevier Ad hoc Networks 2 (1) (2004) 1–22. `doi:10.1016/S1570-8705(03)00043-X`.

[2] A. N. Mian, R. Baldoni, R. Beraldi, A survey of service discovery protocols in multihop mobile ad hoc networks, IEEE Pervasive Computing 8 (1) (2009) 66–74. `doi:10.1109/MPRV.2009.2`.

[3] R. Ahlswede, N. Cai, S.-Y. Li, R. Yeung, Network information flow, IEEE Trans. Information Theory 46 (4) (2000) 1204–1216. `doi:10.1109/18.850663`.

[4] L. Li, R. Ramjee, M. Buddhikot, S. Miller, Network Coding-Based Broadcast in Mobile Ad-hoc Networks, in: Proc. IEEE Int. Conf. Computer Communications (INFOCOM), 2007, pp. 1739–1747. `doi:10.1109/INFCOM.2007.203`.

[5] S. Wang, A. Vasilakos, H. Jiang, X. Ma, W. Liu, K. Peng, B. Liu, Y. Dong, Energy Efficient Broadcasting Using Network Coding Aware Protocol in Wireless Ad Hoc Network, in: IEEE Int. Conf. Communications (ICC), 2011, pp. 1–5. `doi:10.1109/icc.2011.5962913`.

[6] S. Wang, C. Wang, K. Peng, G. Tan, H. Jiang, Y. Dong, Network coding over connected dominating set: energy minimal broadcasting in wireless ad hoc networks, Springer Wireless Networks 20 (5) (2013) 1023–1036. `doi:10.1007/s11276-013-0648-z`.

[7] T. Kunz, L. Li, Broadcasting in multihop mobile tactical networks: To network code or not, in: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, IWCMC '10, ACM, New York, NY, USA, 2010, pp. 676–680. `doi:10.1145/1815396.1815551`.
URL `http://doi.acm.org/10.1145/1815396.1815551`

[8] P. Ostovari, J. Wu, A. Khreishah, Deadline-aware broadcasting in wireless networks with local network coding, in: Proc. IEEE Int. Conf. Computing, Networking and Communications (ICNC), 2012, pp. 1123–1127. `doi:10.1109/ICCNC.2012.6167383`.

[9] S. Yang, J. Wu, Efficient Broadcasting Using Network Coding and Directional Antennas in MANETs, IEEE Trans. Parallel and Distributed Systems 21 (2) (2010) 148–161. `doi:10.1109/TPDS.2009.44`.

[10] C. Fragouli, J. Widmer, J.-Y. Le Boudec, Efficient Broadcasting Using Network Coding, IEEE/ACM Trans. Networking 16 (2) (2008) 450–463. `doi:10.1109/TNET.2007.901080`.

[11] J. Widmer, J.-Y. Le Boudec, Network Coding for Efficient Communication in Extreme Networks, in: Proc. ACM SIGCOMM Workshop Delay-tolerant Networking (WDTN), 2005, pp. 284–291. doi:10.1145/1080139.1080147.

[12] T. Kunz, K. Mahmood, A. Matrawy, Broadcasting in multihop wireless networks: The case for multi-source network coding, in: Proc. IEEE Int. Conf. Communications (ICC), 2012, pp. 6679–6684. doi:10.1109/ICC.2012.6363697.

[13] N. Papanikos, E. Papapetrou, Deterministic broadcasting and random linear network coding in mobile ad hoc networks, Computing Research Repository (CoRR) arXiv:1411.4465.

[14] N. Kadi, K. Agha, Network coding for MPR-based flooding in ad hoc wireless networks under mobility conditions, Springer-Verlag Annals Telecommunications 66 (5-6) (2011) 319–329. doi:10.1007/s12243-010-0207-x.

[15] S. Wang, G. Tan, Y. Liu, H. Jiang, T. He, Coding Opportunity Aware Backbone Metrics for Broadcast in Wireless Networks, IEEE Trans. Parallel and Distributed Systems 25 (8) (2014) 1999–2009. doi:10.1109/TPDS.2013.229.

[16] N. Rahnavard, B. N. Vellambi, F. Fekri, CRBcast: a reliable and energy-efficient broadcast scheme for wireless sensor networks using rateless codes, IEEE Trans. Wireless Communications 7 (12) (2008) 5390–5400. doi:10.1109/T-WC.2008.071218.

[17] B. Vellambi, N. Rahnavard, F. Fekri, FTS: A distributed energy-efficient broadcasting scheme using fountain codes for multihop wireless networks, IEEE Trans. Communications 58 (12) (2010) 3561–3572. doi:10.1109/TCOMM.2010.101210.070495.

[18] A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, M. Zorzi, Toward network coding-based protocols for data broadcasting in wireless Ad Hoc networks, IEEE Trans. Wireless Communications 9 (2) (2010) 662–673. doi:10.1109/TWC.2010.02.081057.

[19] I. Hou, Y. Tsai, T. Abdelzaher, I. Gupta, Adapcode: Adaptive network coding for code updates in wireless sensor networks, in: Proc. IEEE Int. Computer Communications (INFOCOM), 2008, pp. 1517–1525. doi:10.1109/INFOCOM.2008.211.

[20] S. Y. Cho, C. Adjih, Wireless Broadcast with Network Coding: Dynamic Rate Selection, in: Advances in Ad Hoc Networking: Proc. Annual Mediterranean Ad Hoc Networking Workshop, 2008, pp. 191–202. doi:$10.1007/978-0-387-09490-8_{}17$.

[21] Z. Yang, M. Li, W. Lou, R-Code: Network coding-based reliable broadcast in wireless mesh networks, Elsevier Ad Hoc Networks 9 (5) (2011) 788–798. doi:10.1016/j.adhoc.2010.09.009.

[22] J. Subramanian, R. Morris, H. Balakrishnan, UFlood: High-throughput flooding over wireless mesh networks, in: Proc. IEEE Int. Conf. Computer Communications (INFOCOM), 2012, pp. 82–90. doi:10.1109/INFCOM.2012.6195831.

[23] S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading structure for randomness in wireless opportunistic routing, ACM SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 169–180. doi:10.1145/1282427.1282400.

[24] D. Koutsonikolas, Y. C. Hu, C.-C. Wang, Pacifier: High-throughput, Reliable Multicast Without "Crying Babies" in Wireless Mesh Networks, IEEE/ACM Trans. Networking 20 (5) (2012) 1375–1388. doi:10.1109/TNET.2011.2177274.

[25] P. Ostovari, A. Khreishah, J. Wu, Deadline-aware broadcasting in wireless networks with network coding, in: Proc. IEEE Conf. Global Communications (GLOBECOM), 2012, pp. 4435–4440. doi:10.1109/GLOCOM.2012.6503816.

[26] C. Ververidis, G. Polyzos, Service discovery for mobile Ad Hoc networks: a survey of issues and techniques, IEEE Communications Surveys Tutorials 10 (3) (2008) 30–45. doi:10.1109/COMST.2008.4625803.

[27] Y. Luo, O. Wolfson, Mobile P2P Databases, in: Encyclopedia of GIS, Springer US, 2008, pp. 671–677. doi:$10.1007/978-0-387-35973-1_{}795$.

[28] T. Aggelidis, E. Papapetrou, Dominant pruning based query routing in peer databases over manets, in: 2010 IEEE Globecom Workshops, 2010, pp. 1103–1107. doi:10.1109/GLOCOMW.2010.5700106.

[29] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft, Xors in the air: Practical wireless network coding, IEEE/ACM Trans. on Networking 16 (3) (2008) 497–510. doi:10.1109/TNET.2008.923722.

[30] J. Yu, N. Wang, G. Wang, D. Yu, Connected dominating sets in wireless ad hoc and sensor networks – A comprehensive survey, Elsevier Computer Communications 36 (2) (2013) 121–134. doi:10.1016/j.comcom.2012.10.005.

[31] A. Qayyum, L. Viennot, A. Laouiti, Multipoint relaying for flooding broadcast messages in mobile wireless networks, in: Proc. Annu. Hawaii Int. Conf. System Sciences, 2002, pp. 3866–3875. doi:10.1109/HICSS.2002.994521.

[32] H. Lim, C. Kim, Flooding in wireless ad hoc networks, Elsevier Computer Communications 24 (3–4) (2001) 353–363. doi:10.1016/S0140-3664(00)00233-4.

[33] W. Lou, J. Wu, On reducing broadcast redundancy in ad hoc wireless networks, IEEE Trans. Mobile Computing 1 (2) (2002) 111–122. doi:10.1109/TMC.2002.1038347.

[34] B. Williams, T. Camp, Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks, in: ACM Proc. Int. Symp. Mobile Ad Hoc Networking & Computing (MobiHoc), 2002, pp. 194–205. doi:10.1145/513800.513825.

[35] P. Ruiz, P. Bouvry, Survey on Broadcast Algorithms for Mobile Ad Hoc Networks, ACM Comput. Surv. 48 (1) (2015) 8:1–8:35. doi:10.1145/2786005.

[36] T. H. Cormen, C. Stein, R. L. Rivest, C. E. Leiserson, Introduction to Algorithms, 2nd Edition, McGraw-Hill Higher Education, 2001.

[37] M. Baysan, K. Sarac, R. Chandrasekaran, S. Bereg, A polynomial time solution to minimum forwarding set problem in wireless networks under unit disk coverage model, IEEE Trans. Parallel and Distributed Systems 20 (7) (2009) 913–924. doi:10.1109/TPDS.2008.169.

[38] M. Baysan, K. Sarac, R. Chandrasekaran, Polynomial time solution to minimum forwarding set problem in wireless networks under disk coverage model, Elsevier Ad Hoc Networks 10 (7) (2012) 1253 – 1266. doi:10.1016/j.adhoc.2012.03.009.

[39] S. Agathos, E. Papapetrou, On the Set Cover Problem for Broadcasting in Wireless Ad Hoc Networks, IEEE Communications Letters 17 (11) (2013) 2192–2195. doi:10.1109/LCOMM.2013.091913.131765.

[40] W. Woon, K. Yeung, Enhanced Termination Condition for Deterministic Broadcasting Protocols in Mobile Ad Hoc Networks, in: IEEE Vehicular Technology Conference Fall (VTC), 2010, pp. 1–5. doi:10.1109/VETECF.2010.5594507.

[41] L. Zhu, B.-S. Lee, B.-C. Seet, K.-J. Wong, G. Liu, S.-Y. Huang, K.-K. Lee, Performance of New Broadcast Forwarding Criteria in MANET, in: Information Networking. Networking Technologies for Broadband and Mobile Networks, Vol. 3090 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 34–43. doi:$10.1007/978-3-540-25978-7_{}4$.

[42] D. Nguyen, P. Minet, Analysis of MPR Selection in the OLSR Protocol, in: IEEE Int. Conf. Adv. Information Networking and Applications Workshops (AINAW), Vol. 2, 2007, pp. 887–892. doi:10.1109/AINAW.2007.94.

[43] S. McCanne and S. Floyd, ns2 - Network Simulator, http://www.isi.edu/nsnam/ns/.

[44] C. Bettstetter, G. Resta, P. Santi, The node distribution of the random waypoint mobility model for wireless ad hoc networks, IEEE Trans. Mobile Computing 2 (3) (2003) 257–269. doi:10.1109/TMC.2003.1233531.

[45] J. G. Jetcheva, D. B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: ACM Proc. Int. Symp. Mobile Ad Hoc Networking and Computing, 2001, pp. 33–44. doi:10.1145/501422.501423.

[46] J.-Y. Le Boudec, M. Vojnovic, Perfect simulation and stationarity of a class of mobility models, in: Proc. IEEE Int. Conf. Computer Communications (INFOCOM), 2005, pp. 2743–2754. doi:10.1109/INFCOM.2005.1498557.

[47] P. A. Chou, Y. Wu, K. Jain, Practical network coding, in: Proc. Allerton Conf. Communication, Control and Computing, 2003, pp. 40–49.

[48] S.-Y. Li, R. Yeung, N. Cai, Linear network coding, IEEE Trans. Information Theory 49 (2) (2003) 371–381. doi:10.1109/TIT.2002.807285.

[49] T. Ho, R. Koetter, M. Medard, D. Karger, M. Effros, The benefits of coding over routing in a randomized setting, in: IEEE Proc. Int. Symp. Information Theory, 2003, p. 442. doi:10.1109/ISIT.2003.1228459.