Toward Network Coding-Based Protocols for Data Broadcasting in Wireless Ad Hoc Networks

Alfred Asterjadhi, *Student Member, IEEE,* Elena Fasolo, Michele Rossi, *Member, IEEE,*
Joerg Widmer, *Member, IEEE,* and Michele Zorzi, *Fellow, IEEE*

***Abstract*—In this paper we consider practical dissemination algorithms exploiting network coding for data broadcasting in ad hoc wireless networks. For an efficient design, we analyze issues related to the use of network coding in realistic network scenarios. In detail, we quantify the impact of random access schemes, as used by IEEE 802.11, on the performance of network coding. In such scenarios, deadlock situations may occur where the delivery process stops and some of the nodes never gather the required packets. To tackle this problem, we propose a proactive mechanism (called *proactive network coding*) which adapts its transmission schedule according to the decoding status of neighboring nodes. This scheme can detect when nodes need additional packets in order to decode and acts accordingly. We finally investigate the behavior of network coding schemes in multi-rate environments, where we propose a distributed heuristic approach for the selection of data rates.**

***Index Terms*—Wireless ad hoc networks, network coding, data dissemination, protocol design.**

I. Introduction

NETWORK coding is a recently introduced paradigm to efficiently disseminate data in wireless networks, where data flows coming from multiple sources are combined to increase throughput, reduce delay, and enhance robustness. In contrast to the traditional *store* and *forward* approach [1], it implements a *store*, *code*, and *forward* technique, where each node stores incoming packets in its own buffer and transmits their combinations, where combining is performed over some finite Galois field. This technique allows for increased throughput efficiency as well as scalability and robustness [2]. These benefits arise in the case of multicasting [3], [4] as well as for other network configurations, such as multiple unicast communications [5], [6]. Moreover, they are not restricted to error-free communication networks, but can also be obtained in ad hoc networks [7]–[9], peer-to-peer systems [10], and optical networks. Important theoretical results are known, see, e.g., [3], [11]–[14], and research is now moving towards the exploitation of network coding in practical communication protocols. To this end, the work in [3] is of considerable importance as it demonstrates that random linear network coding (RLNC) is able to reach network capacity in practical settings. This is very important as linear random coding is lightweight and inherently localized and, as such, can be exploited by communication protocols at low overhead.

In this paper, we consider the random network coding schemes of [9] for all-to-all data transmission. According to these algorithms, whenever an innovative packet is received at a given node, it generates with probability ρ (the *forwarding factor*) a new packet through RLNC and broadcasts it over the channel. For the medium access control (MAC) we consider several variants of carrier sense multiple access (CSMA). We focus on the interaction between MAC and network coding over different wireless network configurations in order to capture the effects of each protocol component and quantify the performance degradation due to packet collisions and random transmission schedules. Subsequently, based on our performance evaluation we design a *proactive* dissemination mechanism and we show that it outperforms the *reactive* schemes in [9]. The main contributions of this paper are:

- We study the performance of random network coding in selected topologies (circular, grid, and random networks). In particular, we assess the impact of 1) packet collisions and 2) the random mixing of flows as dictated by random access at the MAC layer.
- We identify deadlock situations where the delivery process prematurely stops due to the lack of further innovative packets and, as a result, some of the nodes never fully decode.
- We propose a lightweight and decentralized combination and transmission technique, called proactive network coding, to mitigate the deadlock problem in multi-hop networks.
- We evaluate the impact of the forwarding factor ρ as well as the selection of the transmission rate in multi-rate ad hoc networks.

Related work on communication protocols for wireless networks can be found in [5], [6], [8], [9], [15]–[17]. [15] was the first contribution to present a practical and distributed solution exploiting RLNC. The authors focused on how the coding matrix as well as the information related to the random combination of packets in some finite Galois field $GF(q)$ can be shared by different nodes at low overhead. This is a crucial aspect for network coding algorithms to work in multi-hop radio networks. The scheme of [16] jointly considers packet combinations with ARQ strategies for wireless sensor networks; the transmission paradigm as well as the node constraints are however different from those in wireless ad hoc networks, which is the focus of this paper. COPE [5] applies network coding to unicast flows in wireless networks.

Manuscript received August 8, 2008; revised June 13, 2009, November 24 and 27, 2009; accepted November 27, 2009. The associate editor coordinating the review of this paper and approving it for publication was S. Hanly.

A. Asterjadhi (corresponding author), M. Rossi, and M. Zorzi are with the Department of Information Engineering, DEI, University of Padova, via Gradenigo 6/B – 35131, Padova, Italy (e-mail: aasterja@dei.unipd.it).

E. Fasolo is with IKS s.r.l., Corso Stati Uniti 14 bis – 35127, Padova, Italy.

J. Widmer is with DOCOMO Euro-Labs, Landsberger Strasse 312 – 80687, Munich, Germany.

Digital Object Identifier 10.1109/TWC.2010.02.081057

1536-1276/10$25.00 © 2010 IEEE

The authors of the paper experimentally show that significant gains, in terms of maximum throughput, are possible even in the case of unicast transmissions and even when network coding is implemented through simple XORing of packets within a single-hop neighborhood (rather than forwarding encoded packets over multiple hops). [6] presents BFLY, a localized network coding protocol which recognizes butterfly structures in the network to exploit the coding opportunities they represent. This protocol builds on COPE and also encodes packets through XOR operations, but it additionally allows the transmission of encoded packets over multiple hops. The work in [5], [6] present practical communication schemes, where [5] quantifies the achievable gains from coding when an actual MAC layer is used. Our focus is different in terms of network scenario, as we consider an all-to-all communication paradigm and encode packets in $GF(q)$ with $q = 256$, considering more general coding rules. In [8], the authors investigate the interaction between MAC and network coding in wireless multi-hop networks, and propose distributed and opportunistic scheduling rules for the combination of packets in the presence of time-varying fading links. They also look at the impact of MAC schedules. However, this topic is treated differently from what we do here as packets at relay nodes are XORed and possible deadlocks in the data dissemination are not investigated. The authors of [17] study the interaction of network coding and MAC, devising suitable conflict-free transmission schedules (for a given connectivity graph) and related off- and on-line algorithms for wireless multi-hop networks. However, their strategies entail some coordination among nodes which incurs additional communication overhead with respect to RLNC. Reference [9] studies all-to-all communication scenarios and introduces a class of lightweight *reactive* and distributed network coding protocols based on RLNC, proving the superiority of these schemes over flooding [18] and epidemic routing [1]. Our present work is the natural continuation of [9], which we complement investigating the performance degradation due to actual MAC schemes and proposing a solution to deadlocks in the dissemination of the information, which may occur in certain topologies. The performance evaluation that we carry out in this paper is based on simulation results obtained using ns2; the relevant simulation code can be downloaded from [19].

The remainder of this paper is organized as follows. In Section II we quantify the impact of several IEEE 802.11 variants on the performance of network coding. In Section III we design a lightweight and distributed mechanism to perform network coding more efficiently and show via simulations that it can significantly outperform the data dissemination schemes of [9]. Other results are given for multi-rate environments in Section IV: on the one hand, high data rates are good as they shorten packet transmission times, thereby reducing the collision probability; on the other hand, however, in a multi-hop scenario high data rates mean that packets have to travel more hops to reach the destination. Thus, determining suitable data rates for each node, so as to obtain good tradeoffs between delay and delivery ratio, is a challenging and interesting problem for which we propose a heuristic solution in Section IV. Section V concludes the paper.

II. The Impact of MAC Layer on Data Dissemination via Network Coding

In this section we study the behavior of *Network Coding* in realistic environments. Our aim is to address open issues for its application in wireless networks with a realistic MAC Layer.

Problem description: wireless ad hoc networks are severely constrained by interference and channel impairments, especially in the case of broadcast communication. The use of traditional access mechanisms such as CSMA-like protocols, when multiple nodes transmit, may suffer from a high number of collisions and dropped packets. Two main factors are to be taken into account when using network coding in conjunction with an actual MAC, namely 1) *collisions* and 2) *packet scheduling*. Both collisions and scheduling are the direct consequence of the random (CSMA-like) channel access that we adopt in this study. Collisions impact the performance as fewer packets are collected; as a consequence it takes longer to obtain full rank decoding matrices at the receivers. Packet scheduling refers to the way in which different nodes take turns in transmitting, which is dictated by the MAC rules. The transmission order is important when network coding is used at higher layers as it influences the way encoded packets are created, i.e., which packets are mixed together. In this paper we focus on the analysis of random access schemes as used by IEEE 802.11.

Network topologies: we start our investigation with circular and grid reference scenarios and then consider random topologies. We ensure that all random topologies used in the simulations are connected. To this end, we do a simple breadth-first-search of the underlying connectivity graph and check if all nodes are visited (a standard procedure to check for connectivity). The topology is valid (i.e., it is used in the simulations) if a single connected cluster exists and is discarded otherwise.

Physical layer: for the results in this paper we have implemented an extended version of the ns2 physical layer for IEEE 802.11b/g which includes packet error rate (PER) calculations accounting for modulation, channel effects, and multi-user interference. In detail, the Signal to Interference plus Noise Ratio (SINR) is evaluated for each receiving node and for each packet taking into consideration the interference generated by nearby transmitters. PERs are obtained from precalculated packet error rate curves. For the channel, we use the standard ns2 two ray ground propagation model. More details can be found in [20].

MAC protocols: we consider four different MAC protocols based on CSMA, which is currently the most widely used medium access mechanism in wireless ad hoc networks.

1) IEEE 802.11b: is considered as the baseline MAC. We adopt the basic access provided by IEEE 802.11b that, in the broadcast mode, does not use any acknowledgment mechanism. In case of collision, no retransmission occurs and the packet is lost.

2) IEEE 802.11b with pseudo broadcast [5]: this scheme is an improvement of the basic IEEE 802.11b, where an acknowledgment mechanism is implemented. A given node randomly picks a neighbor and sends a packet to it via unicast by including its address in the packet header. All other neighbors are in promiscuous mode and can thus overhear/decode the transmission. However, only the neighbor which is the intended receiver of the unicast sends back an acknowledgment. This is done according to the basic IEEE 802.11b unicast communication mode (without RTS/CTS). The packet is retransmitted, after a backoff period, in case there is no acknowledgment from the intended receiver. Using this mechanism, only collisions at the addressed receiver can be detected, while collisions occurring at any of the other neighbors are ignored. Also, this strategy does not solve the hidden terminal problem.

3) IEEE 802.11 with pseudo broadcast and RTS/CTS handshake: To further improve the performance we consider the previous scheme with additional RTS/CTS handshake. These control messages are introduced to alleviate the hidden node problem. The CTS is only transmitted by the node addressed in the packet header. As for the previous schemes, this strategy can not detect collisions at all overhearing nodes.

4) Ideal MAC: is a simple mechanism where transmitted packets are only affected by the delay, $\Delta = \ell + \Delta_{\mathrm{prop}} \simeq \ell$, where Δ_{prop} is the propagation delay and ℓ is the packet transmission time. That is, we assume to have an omniscient entity which regulates the transmissions in order to completely avoid interference and collisions. Hence, as a node sends a packet, all its neighbors successfully receive the message after the (fixed) delay Δ. Note that this MAC still schedules packet transmissions according to CSMA and is thus non-ideal from a packet scheduling point of view when network coding is used at the upper layers. This idealized scheme is exploited to pinpoint the effect of packet errors on the performance of network coding.

Network coding: we consider the RLNC paradigm of [3], which is implemented as presented in [15]. At any given node, every new encoded packet is obtained as the linear combination of all packets in its receiving buffer, where the combination coefficients are scalars randomly picked in $GF(q)$. These scalars are then stored in an encoding vector which is sent along with the encoded packet, see [15] for further details. In addition, we implement three different packet combination techniques. The first two are inspired by the work in [9], while the last one is a new proposal. In the following, we detail the packet combination strategies considered in this section. All the presented schemes are characterized by a design parameter, called *forwarding factor* ρ, which is defined as the ratio between the average number of packets transmitted and the average number of innovative packets received, per node. A received packet is innovative whenever it increases the rank of the decoding matrix [9]. For the schemes below ρ, is decided a priori and equal for all nodes.

1) Probabilistic network coding: this approach uses RLNC. Each node sends a random linear combination of the packets in its buffer. When receiving an innovative message a new combination is transmitted with probability ρ (the *forwarding factor*) whereas nothing is transmitted with probability $1 - \rho$. For $\rho = 0.5$, a node on average sends a new message every two innovative packets received. From [21] we know that $\rho = 0.5$ would theoretically (circular topology, ideal scheduling, and no collisions) assure a packet delivery ratio of 1 when the number of neighbors is 2 (the packet delivery ratio is defined in Section II-A).

2) Semi-deterministic network coding: in this case, for a given forwarding factor ρ, each node sends out a new combination after having received exactly $\lceil 1/\rho \rceil$ innovative packets. As an example, $\rho = 0.5$ means that each node deterministically transmits a new combination every two received innovative packets. The forwarding factor, in this case, is not related to a probability, but it is rather used as a threshold on the number of incoming messages.

3) Timed network coding: The two previous schemes have two major drawbacks. The first is that they are particularly sensitive to packet losses, e.g., due to collisions. In fact, if one of the transmitted packets is lost, the propagation of the information through the network could be interrupted. The second drawback is that both probabilistic and semi-deterministic network coding suffer from some inefficiencies when there is a small number of packets to combine. In such cases, new combinations are created from a small set of packets and, for this reason, are often not innovative. To alleviate these problems, we introduce a *timing strategy* into the first scheme. For each received innovative packet, a timer is activated. When the timer has expired, the node decides to send out a new random combination with probability ρ. The timer, τ, is a uniform random variable in $[0, \tau_{\max}]$. This timing approach has two advantages. With the introduction of a waiting interval before transmission, nodes have the chance of collecting other innovative packets and send out richer combinations. Moreover, the reduction of the number of transmissions and the random characteristic of the timer help in decreasing the collision probability at the MAC layer. The drawback of the timed scheme is the introduction of a short delay due to the timer. Hence, the timer value shall be chosen so as to achieve a good trade-off between extra delay and performance improvements. In IEEE 802.11b, this value has to be large enough to allow for the collection of more than one packet, which translates to selecting $\tau_{\max} \approx 10 - 30$ ms. We picked $\tau_{\max} = 20$ ms. Note that in general $\tau_{\max}$ depends on network density and flow demands.

Traffic pattern: for the traffic pattern, each node i inserts into the network a single original packet x_i and wants to collect all the other inserted packets. x_is are generated either *randomly* or *deterministically*. In the former case, each node inserts its original packet by independently picking the insertion time uniformly in a fixed length interval of $\Delta_1 = 100$ ms. In the latter case, we can assume to have a simple application that inserts original packets sequentially in each node. Subsequent insertions, at different nodes, are separated by fixed time intervals of $\Delta_2 = 1$ s. For this value of Δ_2, with the considered scenario (e.g., transmission times and network size), the collision probability is negligible for both original packets and subsequent transmissions elicited by network coding. This is useful to assess the performance of our dissemination schemes when used with an ideal MAC.

A. MAC and Combination Strategies: Simulation results

In this section, we discuss the most relevant results we obtained via ns2 simulations. All presented schemes are eval-

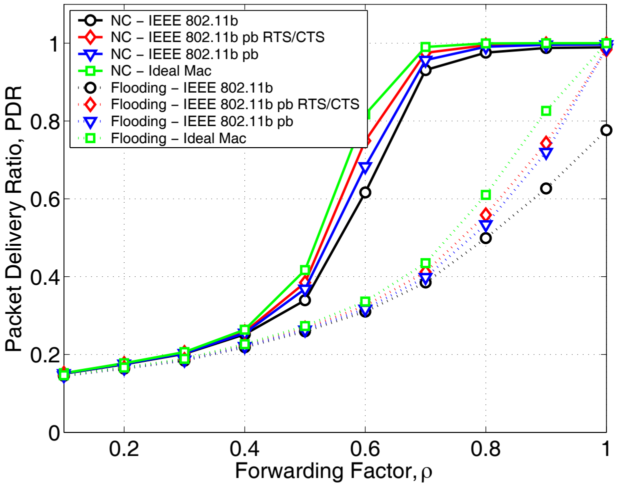

Fig. 1. Packet Delivery Ratio: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in circular networks with $n = 16$.

uated taking into account the *random* and the *deterministic* traffic patterns. We test the algorithms varying ρ from 0.1 to 1 and the number of nodes in the network, n, from 4 to 64, and we consider a data rate of $R = 1$ Mbps. For all MACs we consider packet lengths of $L = PHY + MAC + NC(n) + P$, where PHY and MAC are the physical and the MAC headers, respectively, with $PHY = 192$ bits, $MAC = 224$ bits. $NC(n)$ is the network coding header that depends on the number of nodes n and is of size $NC(n) = 8(18 + n)$ bits. P is the payload size that in our case is 64 bits.[1] We organize our performance analysis in two parts: we first focus on the impact of different MAC protocols on network coding and we subsequently evaluate the effect of different packet combination strategies.

Our performance metrics are: 1) the *Packet Delivery Ratio,* PDR, which is defined as the ratio between the number of successfully received (and decoded) packets and the number of packets a node is interested in, averaged over all nodes; 2) *Packet Delivery Delay:* is the average time between the first transmission of a packet and its reception and successful decoding at the destination nodes. This metric is only computed for correctly received packets; 3) *Protocol Overhead:* is the ratio between the number of transmitted packets at the MAC layer and the number of successfully decoded packets. This value depends on the adopted MAC protocol and on the efficiency of the network coding strategy.

Impact of MAC protocols: in Fig. 1 we compare probabilistic network coding (solid lines) against probabilistic flooding (dotted lines) in terms of delivery ratio in a circular network topology for $n = 16$. Network coding outperforms probabilistic flooding for all values of ρ. Gains are more pronounced when ρ is close to one and n is large (results for different values of n are not shown here due to space constraints, the results are however similar to those in [9]). As observed in [21], for this topology a delivery ratio of one is theoretically achievable with ρ slightly larger than 0.5.

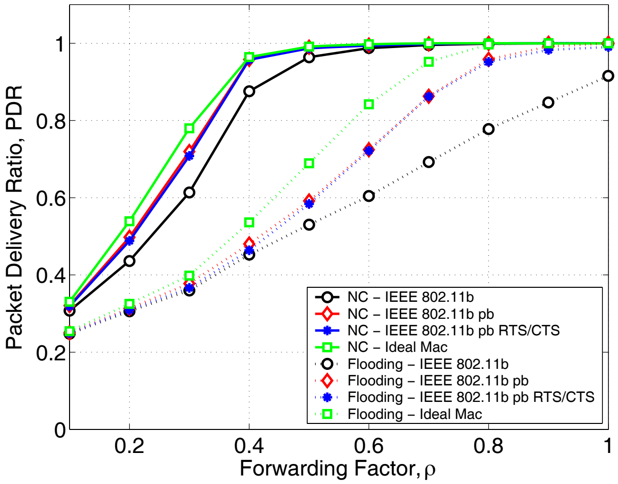

Fig. 2. Packet Delivery Ratio: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with $n = 16$.

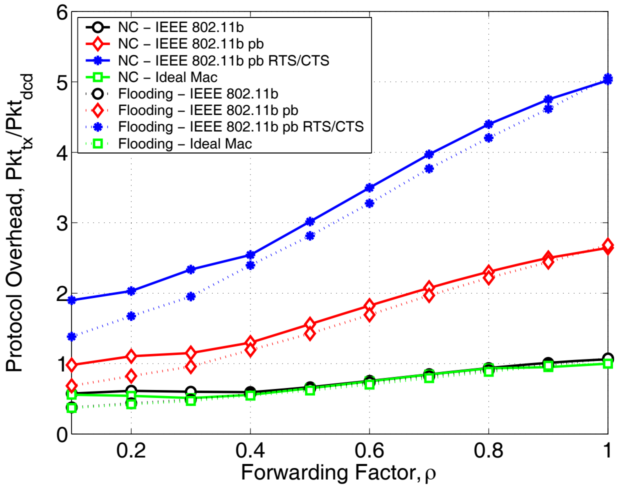

Fig. 3. Protocol Overhead: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with $n = 16$.

This is obtained through a proper centralized coordination of the nodes' transmissions which maximizes the probability of sending innovative packets at each transmission attempt. However, this performance level is never reached in practice and the actual PDR depends on the number of nodes. Looking at Fig. 1 for $\rho = 0.6$, IEEE 802.11b achieves $PDR \approx 0.6$, whereas an ideal MAC achieves $PDR \approx 0.8$, which corresponds to a decrease in performance of about 25%. Note that our ideal MAC does not provide full reliability as it still schedules transmissions according to CSMA and does not use the optimal coordination strategy of [21]. The effectiveness of pseudo broadcast (IEEE 802.11 pb in the figure) and pseudo broadcast with RTS/CTS (IEEE 802.11 pb RTS/CTS) is also clear, though the improvements are not as large as expected. The observed decrease in performance is due to the use of an actual MAC layer (IEEE 802.11b in this case) and to the sub–optimality of random scheduling, which indicates the importance of these issues for the design of practical schemes.

Fig. 2 shows results for a different setting where $n = 16$ nodes are placed over a grid. As expected, the achieved

[1] Due to the inefficiencies of 802.11 medium access and the additional network coding overhead, using a small packet size represents a lower bound on throughput performance. The relative performance differences between the protocols remain unchanged when using larger packet sizes.

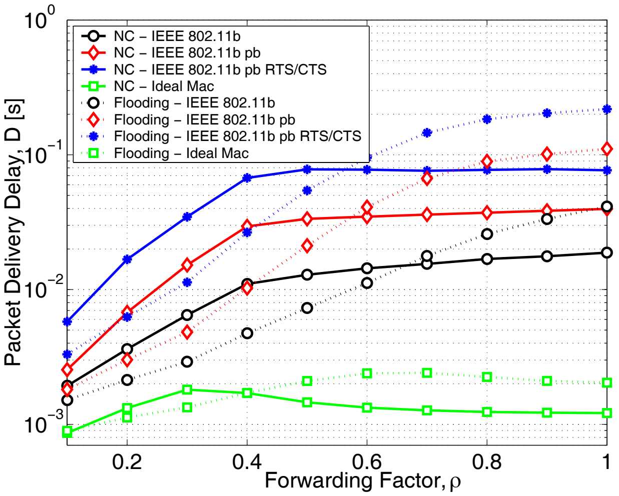

Fig. 4. Packet Delivery Delay: Performance comparison of *Probabilistic network coding* and *Probabilistic Flooding* for different MAC protocols in grid networks with $n = 16$.

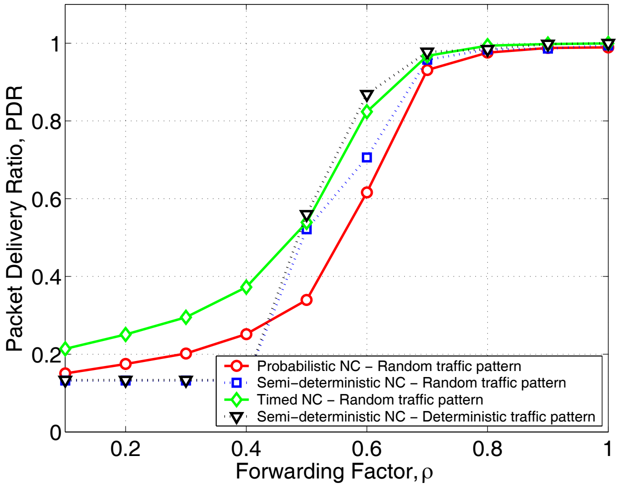

Fig. 5. Performance comparison of different combination strategies in circular networks with $n = 16$ and IEEE 802.11 MAC.

performance is better than in the circular case due to the higher number of neighbors per node (4 instead of 2), which favors packet mixing and dissemination. Also in this scenario, the presence of realistic MAC layers significantly reduces the PDR metric for a given ρ. As expected, the schemes implementing collision avoidance policies (i.e., IEEE 802.11b with pseudo broadcast and IEEE 802.11 with pseudo broadcast and RTS/CTS handshake) improve PDR but also increase the protocol overhead. This is due to the MAC retransmissions in case of collisions and to the control traffic (i.e., ACK, RTS and CTS packets). In addition, we note that when we compare the performance of probabilistic network coding and flooding against ρ, we have a fair comparison as, given a specific ρ and a fixed MAC protocol, both network coding and flooding lead to very similar protocol overhead (see Fig. 3). Pseudo broadcast and pseudo broadcast with RTS/CTS are effective in decreasing the number of collisions. However, using these additional techniques to recover from packet loss leads to longer delays, as can be seen from Fig. 4. The average delay increase is about one order of magnitude in the worst case (pseudo broadcast with RTS/CTS). We also note that the delay of network coding stabilizes for increasing ρ while it continues to increase for flooding. The reason for this is that with flooding, a higher number of redundant packets is received early on, delaying the reception of innovative packets. For network coding, the combination of packets prevents this from happening and most packets received are innovative even for high ρ.

To sum up, we observe that the presence of actual MAC protocols reduces the performance in terms of packet delivery ratio; however, this reduction is not as high as expected. In addition, collision avoidance policies give little improvement in terms of PDR, while leading to poor overhead and delay performance.

Impact of packet combination strategies: Fig. 5 shows the packet delivery ratio performance for a circular network with $n = 16$ for various packet combination strategies for an IEEE 802.11b MAC protocol. The semi-deterministic schemes (dotted lines) show a phase change, where PDR remains constant up to $\rho^* = 0.4$ and then suddenly increases for higher forwarding factors. This does not occur for probabilistic network coding (solid lines) whose curves are smooth. This reflects the threshold based transmission policy of semi-deterministic network coding. The exact value of the shifting point ρ^* depends on the number of neighbors. For circular networks, where each node has exactly two neighbors, $\rho < 0.5$ ($\lceil 1/\rho \rceil > 2$) never suffices to trigger the transmission of a new combination, as the initial number of innovative packets is equal to two. This flaw is not present in probabilistic and timed network coding, where sending rules are based on probabilities rather than on hard thresholds. Timed network coding outperforms the semi-deterministic scheme with deterministic traffic pattern for $\rho \leq \rho^*$ and performs very close to this method for larger forwarding factors. In addition, the timed strategy performs better than both semi-deterministic and probabilistic network coding with random scheduling. For $\rho = 0.5$, probabilistic network coding with random scheduling achieves $PDR \approx 0.35$, whereas timed network coding leads to $PDR \approx 0.55$, which corresponds to an improvement of about 57%. We observed that the timed strategy introduces an additional delay. Also, there are some expected differences between ideal and actual MAC. For IEEE 802.11b, the delay increase is reasonably small (approximately equal to the average value of the timer) and is similar to that introduced by the pseudo broadcast algorithms. Hence, the timed combination provides higher benefits than pseudo broadcast in terms of packet delivery ratio, leading to similar extra delays. For this reason, the timed scheme may make sense when the goal is to maximize the packet delivery ratio (throughput) while accepting some delay degradation. This extra delay appears, however, well tolerable (less than 5% increase over the delay without the timed strategy). The results of probabilistic and timed network coding for random topologies are plotted in Fig. 6 (similar results hold for grid networks). Note that in these networks the gain becomes larger for decreasing ρ; in fact, when fewer packets are transmitted timed network coding more effectively exploits coding opportunities.

From the above results we see that there exists a gap between the predicted theoretical performance and the results

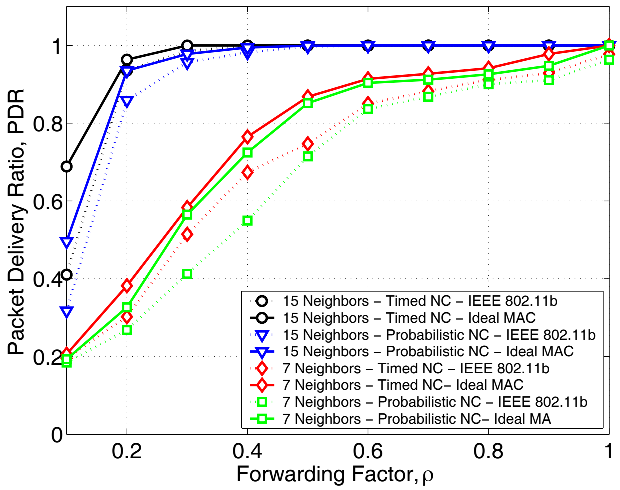

Fig. 6. Random Networks: Performance comparison of Packet Delivery Ratio for *Probabilistic* and *Timed Network Coding* for ideal and IEEE 802.11b MAC.

obtained in realistic environments. This gap is due to the random access mechanism of IEEE 802.11, which leads to collisions, and also to its packet scheduling, which does not allow the combination of packets in the optimal order.

III. Proactive Network Coding (ProNC)

The schemes considered up to now are *reactive* protocols, i.e., nodes participate in the dissemination of data only when they receive innovative information. If this does not occur, the dissemination is interrupted even though nodes may still have innovative information to send. This fact is an inherent characteristic of the reactive approach. In this section we propose a network coding data dissemination scheme based on a *proactive* approach (referred to in the following as *ProNC*) to address this problem. Even though our focus here is on scenarios where data is to be exchanged among all the users of a wireless ad hoc network, the rationale behind *ProNC* also applies to different settings. Our scheme is completely distributed and self-adaptable and requires very limited network knowledge, which can be easily acquired by overhearing the exchanged data.

In the previous section we have seen that reactive schemes are likely to suffer from the presence of interference and collisions in realistic radio environments. The main problem of reactive schemes is that new random combinations are generated and transmitted only when innovative (i.e., linearly independent) information is received. Innovative packets may however be lost due to packet collisions, thus interrupting data propagation. Even worse, the insertion of innovative information into a given network area often causes all nodes in the area to attempt their new transmissions simultaneously and this further increases the collision probability.

In reactive probabilistic network coding, nodes send out new combinations based on a forwarding factor ρ, which depends on their number of neighbors [9]. Setting ρ inversely proportional to the number of neighbors has the desirable effect that the number of innovative packets per area is independent of the local node density. We observe that there are particular topologies where this strategy does not work. As an example, think of the case where a given node x has a large number of neighbors and one of them, say node y, has only x as its neighbor. Due to its high number of neighbors (small ρ), x sends out a small number of packets and, in turn, y is unlikely to be able to decode all the wanted information (as it did not receive enough independent combinations from x). In contrast, our *proactive* scheme does not require the reception of innovative information to continue data dissemination (so it is more robust to interference and collisions), and its performance does not depend on the forwarding factor ρ. It is based on two important components: 1) a set of conditions to stop transmissions when all original packets have been delivered to all nodes, i.e., *Stopping Conditions (SC)* and 2) a strategy to set the frequency at which new random packet combinations are to be sent so as to avoid network congestion. In the rest of the section we refer to this strategy as *Rate Adaptation* mechanism.

Basic rules for ProNC: each node can be in one of two different states: *active* and *inactive*. The basic idea of the proactive approach is that an *active* node periodically sends out a new packet combination to its neighbors, while an *inactive* node does not transmit. To switch from one state to the other, a node considers the following set of rules:

R1 *A node becomes active upon receiving the first innovative packet.* This means that a data dissemination phase is started and the node has to contribute to it.

R2 *A node becomes inactive when the Stopping Condition is verified.* In this case, further transmissions from this node are no longer useful for its neighbors and should be suppressed to avoid unnecessary overhead.

R3 *A node becomes active again when the Stopping Condition no longer holds.* This last rule is particularly important as it allows propagation of new information into an area where all nodes are currently inactive.

Note that while a node is inactive, it can still receive packets from its neighbors. This information is used to assess whether the stopping condition still holds.

Stopping conditions: there are different ways to define the *Stopping Conditions* for proactive network coding. They depend, in general, on the amount of information that each node has to collect in order to decide whether to suspend its transmissions. Our main aim is to keep the transmission overhead as low as possible. We identify two simple cases in which a node has to suspend its transmission: 1) In the first case, all neighbors of a node x have decoded all the packets they require and thus no further transmissions by x are necessary. 2) The second is when the subspace spanned by the information vectors (i.e., packets) available at node x is contained in the subspace spanned by the information vectors at each of the node's neighbors. In this case, x's packets will not be innovative for any of its neighbors and the node should suspend its transmission.

Based on these observations, we propose two different conditions which are referred to as *Strong* and *Weak* Stopping Conditions (*SSC* and *WSC*, respectively). They define two different proactive schemes. According to the *SSC*, nodes send out beacons (*Strong Stopping Messages, SSM*) to their neighbors when they have decoded all the packets they are interested in. Each node collects *SSM*s from its neighborhood.

When a node receives an *SSM* from each of its known neighbors, the *SSC* is verified and transmissions are stopped. We refer to this scheme as *Strong ProNC* as it requires strong assumptions on the data traffic. In order to send out *SSM*s, each node needs to know in advance how many packets it wants to collect. This fact implies that each node has full knowledge about the amount (and type) of data flowing over the network. Note that the collection of this information may not be feasible in practice.

The second strategy we propose is based on the *WSC*. During data propagation, each node sends out beacons (*Weak Stopping Messages, WSM*) containing a *decoding field* which is set to 1 if it can decode all packets in its buffer and to 0 otherwise. In addition, beacons contain a *rank field* specifying the rank of the nodes' decoding matrices. According to the *Weak Stopping Conditions* each node suspends its transmissions when all its neighbors can decode all the packets in their buffers and their matrices have the same (full) rank. We refer to this second strategy as *Weak ProNC* because it does not require any knowledge about the data traffic and has a limited overhead. However, *Weak ProNC* is suboptimal as there are some situations in which the rank alone does not capture the exact decoding status at different nodes. For instance, it might happen that all neighbors of a node can decode all the packets in their buffers and they all have the same rank but the decoded information is different.

A. Rate adaptation heuristics

We define τ as the time elapsed between the completion of the transmission of a packet by the PHY and the instant when the next packet is made available for transmission at the MAC, i.e., the idle time of the node. Note that τ is (roughly) inversely proportional to the transmission rate of the nodes. In what follows, we present an approximate model to find the value of τ that maximizes the amount of innovative information that is transferred over the channel as a function of the system parameters. Note that the physical (PHY) layer data rate is kept constant. Methods to change the PHY rate are investigated in Section IV.

Impact of MAC layer dynamics: in what follows we derive the relationship between the value of $\tau_{\text{avg}} = E[\tau]$ that maximizes the throughput, referred to as $\tau^{\star}_{\text{avg}}$, and the number of neighbors at any given node, n_v. We consider the packet transmission process in a given neighborhood of $n = n_v + 1$ nodes making the following assumptions: A1) we neglect the channel propagation delay as for the considered system parameters it has a negligible impact on the throughput performance, A2) we assume that all packets involved in a collision are lost and A3) we assume that any transmitted packet is always successfully received by all nodes in the neighborhood unless it collides with another transmission. To obtain a rate adaptation heuristic we model the IEEE 802.11 broadcast communication process. As in [22], the evolution of the transmission backoff counter is tracked using a suitable Markov chain. However, in our case the backoff window size W is always constant as packets are neither retransmitted nor acknowledged. This implies that the broadcast backoff process of any of the $n_v + 1$ nodes can be modeled through the one dimensional Markov chain of Fig. 7 (a two dimensional chain was used in [22] for the IEEE 802.11 unicast case to take retransmissions into account). In addition, in order to model the transmission process in nonsaturated traffic conditions, we adopt the technique of [23] where idle transmission times (due to empty transmission queues) are modeled through the addition of the further state -1. With reference to Fig. 7 our model works as follows. When a target node has a packet to transmit, it starts the backoff process by randomly selecting a backoff counter value from 0 to $W-1$ and then starting to decrement the counter until state 0 is reached. State $i = 0, 1, \ldots, W-1$ represents the current backoff counter value. The transition from state i to state $i-1$ occurs with probability $1-p$ after a backoff slot time (of fixed duration σ), while with probability p the process remains in state i. p is the probability that at least one of the remaining n_v nodes transmit when the target node is in state i. If this occurs, the node momentarily stops counting down its backoff timer. The current packet is finally transmitted when the backoff process reaches state 0 (*transmission state*). Upon the completion of the packet transmission two events can occur: E1) with probability q the transmission queue is non-empty and a new backoff timer is uniformly selected at random in $\{0, 1, \ldots, W-1\}$: the probability that the system moves from state 0 to any state $i = 0, 1, \ldots, W-1$ is thus q/W. E2) With probability $1-q$ the transmission queue is empty and in this case the process moves to the idle state -1, where it remains until a new packet arrives (at which point the chain evolution is the same as in E1).

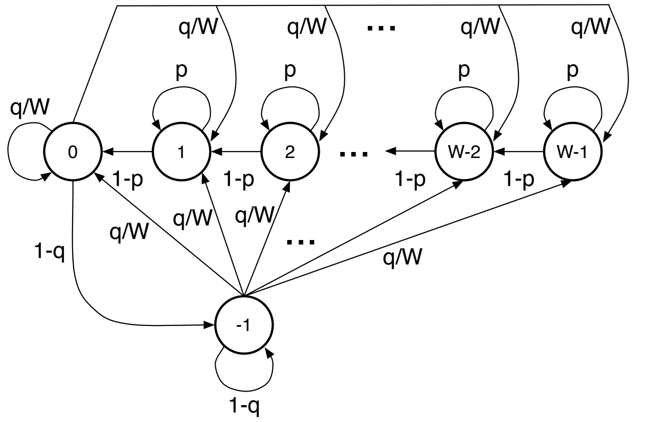

Fig. 7. Markov chain tracking the evolution of the IEEE 802.11 broadcast transmission process. For n_v nodes in a neighborhood, p is the probability that at least one of the remaining n_v nodes transmits when the target node is in slot $i = 1, \ldots, W-1$. $1-q$ is the probability that the transmission buffer is empty after the transmission of a given packet. W is the backoff window size of IEEE 802.11.

Let π_i, $i = -1, 0, 1, \ldots, W-1$ be the steady-state probabilities of the above Markov chain. Our goal is to find π_0 (the *transmission probability*) and relate it to τ_{avg}. From the chain regularities, and by computing recursively through the chain from right to left, we obtain:

$$\begin{aligned} \pi_{W-i} &= \frac{iq(\pi_0 + \pi_{-1})}{W(1-p)}\ , \ i = 1, 2, \ldots, W-1\ , \\ \pi_0 &= (\pi_0 + \pi_{-1})q\ . \end{aligned} \tag{1}$$

From the normalization condition $\sum_{i=-1}^{W-1} \pi_i = 1$ we find

$$\pi_0 + \pi_{-1} = \frac{1}{1 + \frac{(W-1)q}{2(1-p)}} \,, \tag{2}$$

from which we finally obtain π_0 as:

$$\pi_0 = \frac{q}{1 + \frac{(W-1)q}{2(1-p)}} \,. \tag{3}$$

The probability p that at least one of the remaining n_v nodes transmits when the target node is in slot $i = 1, \ldots, W-1$ is found as: $p = 1 - (1-\pi_0)^{n_v} \overset{def}{=} f_1(\pi_0)$. We additionally define the probability P_t that at least one node is transmitting in a given slot,

$$P_t = 1 - (1-\pi_0)^{n_v+1} \,, \tag{4}$$

and P_s as the probability that only one node is in the transmission state, conditioned on the fact that at least one node is transmitting. P_s is obtained as:

$$P_s = \frac{(n_v+1)\pi_0(1-\pi_0)^{n_v}}{P_t} = \frac{(n_v+1)\pi_0(1-\pi_0)^{n_v}}{1-(1-\pi_0)^{n_v+1}} \,. \tag{5}$$

We are now ready to calculate the normalized throughput S as (see [22]):

$$S = \frac{E[\text{payload bits successfully transmitted in a slot}]}{E[\text{slot length}]} \,. \tag{6}$$

Since a successful transmission occurs in a slot with probability $P_t P_s$, the average number of payload bits successfully transmitted in a slot time is $P_t P_s P$, where P is the payload size. The average length of a slot can be obtained by considering the following three mutually exclusive cases: C1) with probability $1 - P_t$, none of the nodes transmit in the slot. The duration of an empty backoff slot is $\sigma = 20$ μs. C2) with probability $P_t P_s$, only one node transmits in the slot. The slot duration in this case is T_s, which is the transmission time of a packet, given by the sum of the time spent transmitting the physical header (T_{PHY}), the MAC header (T_{MAC}), the network coding header ($T_{\text{NC}}(n) = NC(n)/R$), the payload ($T_P = P/R$), plus the distributed inter frame space (DIFS) time, T_{DIFS}:

$$T_s = T_{\text{PHY}} + T_{\text{MAC}} + T_{\text{NC}}(n) + T_P + T_{\text{DIFS}} \,. \tag{7}$$

C3) With probability $P_t(1 - P_s)$, multiple nodes transmit in a slot time, leading to a collision event. The duration of this slot is also T_s because packets are not acknowledged and, therefore, the transmission period for successful and collided packets is the same. Hence, we have $E[\text{slot length}] = \sigma(1 - P_t) + P_t P_s T_s + P_t(1-P_s)T_s$. These facts together with (6) give:

$$\begin{aligned} S &= \frac{P_t P_s P}{\sigma(1-P_t) + P_t P_s T_s + P_t(1-P_s)T_s} \\ &= \frac{P_s(P/\sigma)}{(1-P_t)/P_t + T_s/\sigma} \,. \end{aligned} \tag{8}$$

Note that the maximum throughput is achieved when the following function is maximized:

$$\begin{aligned} g(\pi_0) &= \frac{P_s}{(1-P_t)/P_t + T_s/\sigma} \\ &= \frac{(n_v+1)\pi_0(1-\pi_0)^{n_v}}{T_s/\sigma - (1-\pi_0)^{n_v+1}(T_s/\sigma - 1)} \,. \end{aligned} \tag{9}$$

The optimal transmission probability $\pi_0^\star$ can be found as (see calculations in Section VI of [22]):

$$\begin{aligned} \pi_0^\star &= \operatorname*{argmax}_{\pi_0} g(\pi_0) \\ &= \frac{\sqrt{[(n_v+1) + 2n_v(T_s/\sigma - 1)]/(n_v+1)} - 1}{n_v(T_s/\sigma - 1)} \end{aligned} \tag{10}$$

The optimal $\tau_{\text{avg}}^\star$, is obtained from $\pi_0^\star$ as:

1. Express q as $q = 1 - e^{-E[\text{slot length}]/\tau_{\text{avg}}}$, which as shown in [23] provides a good approximation of the queue behavior in the unsaturated case. Inverting this relation gives:

$$\begin{aligned} \tau_{\text{avg}} &= -\frac{E[\text{slot length}]}{\log(1-q)} \\ &= -\frac{\sigma(1-P_t) + P_t T_s}{\log(1-q)} \,. \end{aligned} \tag{11}$$

2. Obtain $\pi_0^\star$ from (10).
3. Invert (3) to find p as a function of π_0 as:

$$p = 1 - \frac{\pi_0(W-1)}{2(1-\pi_0/q)} \overset{def}{=} f_2(\pi_0, q) \tag{12}$$

and find $q^\star$ as the solution of $f_1(\pi_0^\star) - f_2(\pi_0^\star, q) = 0$, which leads to:

$$q^\star = \frac{2\pi_0^\star(1-\pi_0^\star)^{n_v}}{2(1-\pi_0^\star)^{n_v} + \pi_0^\star(1-W)} \,. \tag{13}$$

4. Obtain $\tau_{\text{avg}}^\star$ from (11) setting $q \leftarrow q^\star$ and expanding $E[\text{slot length}]$ using (4):

$$\tau_{\text{avg}}^\star = \frac{T_s - (1-\pi_0^\star)^{n_v+1}(T_s - \sigma)}{\log\left(\frac{2(1-\pi_0^\star)^{n_v} + \pi_0^\star(1-W)}{2(1-\pi_0^\star)^{n_v+1} + \pi_0^\star(1-W)}\right)} \tag{14}$$

We observe that our model is accurate for sufficiently large n_v, i.e., $n_v \geq 4$. For smaller values it is however inaccurate because of several approximations made in the analysis, i.e., the expression of q [23], the independence of the busy channel probability p among subsequent access slots [22] and to the fact that in our derivation of the steady state probabilities we neglect the semi-Markov character of the process, i.e., that states 0 and $i \neq 0$ have different durations (a common simplification for the analysis of IEEE 802.11 throughput [22], [23]).

Implementation notes: in our implementation we pick $\tau_{\text{avg}} = \tau_{\text{avg}}^\star$ selecting τ uniformly in $[0, 2\tau_{\text{avg}}]$, which gives $E[\tau] = \tau_{\text{avg}}$. We obtain $\tau_{\text{avg}}^\star$ as a function of n_v from (14) using $T_{\text{PHY}} = 192$ μs, $T_{\text{MAC}} = 224$ μs, $T_{\text{DIFS}} = 50$ μs, $\sigma = 20$ μs, $R = 1$ Mbps and $W = 32$ slots, which are used for IEEE 802.11b broadcast with a rate of 1 Mbps. We find that the relationship between the two is well approximated by a linear function, as predicted by the simulation results of [9], [24]. A good approximation is in fact given by the following heuristic: $\tau_{\text{avg}}^\star \simeq \kappa n_v T_s$, where $\kappa = 0.7$. We note that τ_{avg} corresponds to the average amount of time spent in state -1, i.e., to the time elapsed between the completion of the transmission of a packet by the PHY and the instant when a new packet is made available by *ProNC*. The inter-packet transmission time is greater than τ_{avg} as it also includes the time spent in backoff. Moreover, *ProNC* requires the estimation of the number of neighbors at each node which

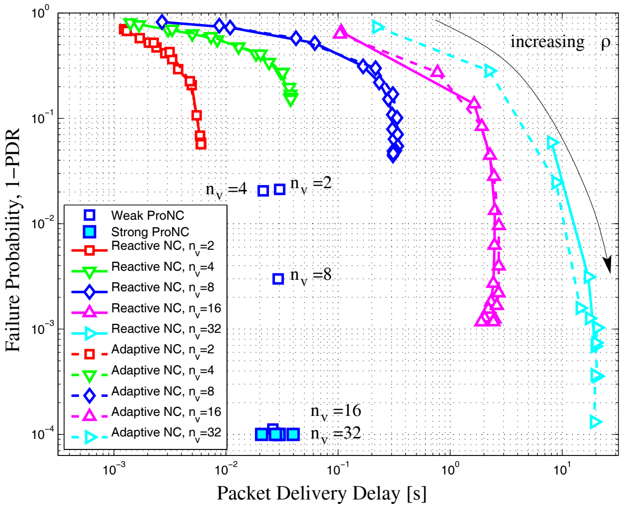

Fig. 8. Failure probability $1-PDR$ vs delay: comparison between proactive and reactive schemes. The curves shown for reactive schemes are obtained for different values of $n_v \in \{2,4,8,16,32\}$, varying ρ as the independent parameter.

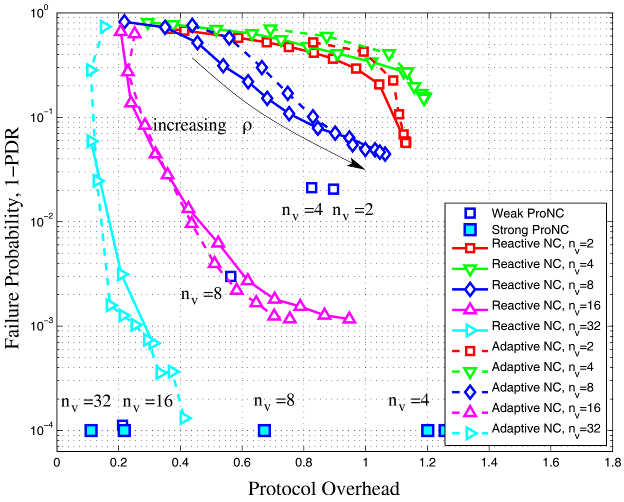

Fig. 9. Failure probability $1-PDR$ vs overhead: comparison between proactive and reactive schemes. The curves shown for reactive schemes are obtained for different values of $n_v \in \{2,4,8,16,32\}$, varying ρ as the independent parameter.

can be simply achieved by monitoring the source addresses of incoming packets. Note that both the stopping conditions and the packet rate adaptation mechanism depend on the node density. Most importantly, the number of nodes estimated in this way is smaller than that of the previous analysis. In fact, this number of neighbors only accounts for the nodes within transmission range, whereas n_v should include all nodes in the collision domain, whose range is always greater. Hence, the actual inter-packet transmission time τ'_{avg}, which accounts for all these facts, is $\tau'_{\text{avg}} = \kappa' n_v T_s$ where $\kappa' = 6$. We thus use this linear heuristic which gives good results across all simulations. In addition, *Stopping Messages* are included within data packets at the cost of a few extra bits. For *SSM*, we need one additional bit, whereas for *WSM* we need a bit to represent the decoding status and a byte to communicate the rank of the local decoding matrix[2]. In both cases, the additional overhead is acceptable. On the downside, when a node becomes inactive it must send out at least one *Stopping Message* to communicate its change of status and this packet may be useless for coding purposes. We note that piggybacking control information within data packets has the beneficial effect of keeping channel congestion low. In addition, the added control information (*SSM*s and *WSM*s, rank, decoding status) is used to increase the efficiency of network coding schemes which, in turn, can further reduce the number of transmissions for a target performance level. These benefits are quantitatively verified below.

B. ProNC*: Simulation Results*

Next, we compare the *ProNC* scheme against the reactive probabilistic schemes proposed in [9]. The results that follow are for topologies where nodes are randomly placed within a fixed area in such a way that the topology is always connected, possibly through multi-hop paths. We consider several average node densities by varying the average number of neighbors, $n_v \in \{2,4,8,16,32\}$. For the MAC, we adopt the basic IEEE802.11b broadcast mode, accounting for channel errors and collisions. In the following graphs we consider the probabilistic network coding of Section II (referred to as ”Reactive NC” in the plots) as well as the adaptive network coding (referred to as ”Adaptive NC”) scheme of [9] where ρ is picked independently at each node as $\rho = c/(n_v+1)$, where c is a suitable constant equal for all nodes [9]. In Fig. 8 we show the tradeoff between failure probability, $1-PDR$, and the packet delivery delay. We note that *ProNC* performs better in terms of data recovery; $1-PDR$ is at least one order of magnitude smaller for *ProNC* when $n_v \in \{8,16\}$. For small n_v, i.e., $n_v \in \{2,4\}$, we often obtain pathological topologies leading to deadlocks of the data dissemination when reactive protocols are used. *ProNC* efficiently deals with these topologies and alleviates the deadlock problem by offering better performance in terms of $1-PDR$. On the downside, in these cases the dissemination of data in *ProNC* takes slightly longer due to the waiting periods of proactive schemes (see variable τ). The tradeoff concerning the protocol overhead is shown in Fig. 9: similarly to reactive schemes, the overhead of *ProNC* increases with decreasing n_v. This is because network coding is more efficient when the node density is high. Also, the overhead of *ProNC* is usually smaller than that of reactive schemes, while it always outperforms reactive solutions in terms of data recovery performance. For both graphs *Weak ProNC* performs slightly worse than *Strong ProNC* in terms of packet delivery ratio, whereas it performs better in terms of overhead performance for the same n_v. The difference in performance is more significant at small densities, i.e., where deadlocks are more likely to occur. As demonstrated in [25], the overhead performance of all schemes at high densities approaches that of an idealized scheme, having the minimum possible overhead: this reflects the fact that network coding works better when there are more coding opportunities.

To summarize, both *Strong* and *Weak ProNC* show satisfac-

[2] A single byte often suffices in practice, i.e., the number of nodes in the network that generate original packets is lower than or equal to 256. Coding over more original packets would imply the inversion of large matrices at the receiver which is impractical and difficult to obtain in realtime.

tory performance in actual network settings. *Weak ProNC* is a distributed and self-adaptable dissemination protocol which does not require any knowledge about the traffic and only requires a few local interactions among nodes to work properly. Further improvements of *Weak ProNC* in terms of PDR are possible through the extension of the communication of control messages over multiple hops. These issues are left for future research.

IV. Network Coding in Multi-Rate Ad Hoc Networks

When nodes have multi-rate transmission capabilities, it is necessary to pick a suitable PHY layer data rate, besides the selection of the forwarding factor. This is not an easy task. When nodes transmit at low data rates their coverage area is larger and in this case packets travel long hops. Conversely, transmitting with higher data rates leads to shorter transmission delays and shorter hops. Whichever is best depends on several factors such as network density and topology. The analysis in, e.g., [26] can be used to calculate the involved delay-throughput tradeoffs for different rates. The objective of the following paragraphs is to obtain good tradeoffs between PDR and average delay in IEEE 802.11g-based multi-rate scenarios in the presence of RLNC.

As in Section III-B, we consider a random wireless network where nodes want to disseminate and retrieve information through multi-rate network coding. We investigate the cooperation between IEEE 802.11g PHY/MAC and network coding in this random scenario considering the all-to-all transmission paradigm. We present results obtained through ns2 simulations with varying forwarding factor ρ for reactive network coding. As per our discussion above, ρ is the probability of sending a new combination whenever a node receives an innovative packet and directly determines the amount of traffic which flows through the wireless network. Note that *ProNC* can be applied to a multi-rate transmission scenario as well and, for this case, we found similar advantages as those presented in Section III-B.

A. Rate Adaptation Heuristics and Simulation Results

In the following, we present a data rate adaptation heuristic which tries to achieve, at the same time, a short delay and a high PDR. We assume that wireless nodes initially have no knowledge about the network status, i.e., they are completely unaware of location and number of neighbors as well as the transmission opportunities in their neighborhood. In order to acquire this knowledge the algorithm uses internal variables, at the MAC layer, to store auxiliary pieces of information such as node addresses, signal to interference plus noise ratios, $SINR$, as well as data rates.

Initially, all nodes begin their transmissions with the lowest available rate (which is 6 Mbps for IEEE 802.11g). This is the best choice in terms of neighbor discovery as it allows nodes to collect information from a larger area. The subsequent reception of packets permits the gathering of useful information which will determine the data rate at this node, as we explain next. Each time a node, say node i, receives a new packet from one of its neighbors j, it extracts the following information: 1) $\mathtt{id}$: the address of the sending node, 2) L: the size of the received packet, 3) γ_j: the instantaneous $SINR$ associated with this packet reception. Upon reception, an average $SINR$, $\overline{\gamma}_j$, is updated for each neighbor j according to a discrete time first order low pass filter as: $\overline{\gamma}_j = \alpha\gamma_j + (1-\alpha)\overline{\gamma}_j$, where $\alpha \in (0,1)$ is the smoothing factor. For our simulations we picked $\alpha = 0.5$. In this way we take into account the variations in the received SINRs, while trying to capture its average value. The packet transmission length L and $\overline{\gamma}_j$ are thus used to estimate the PER for neighbor j for all transmission rates.

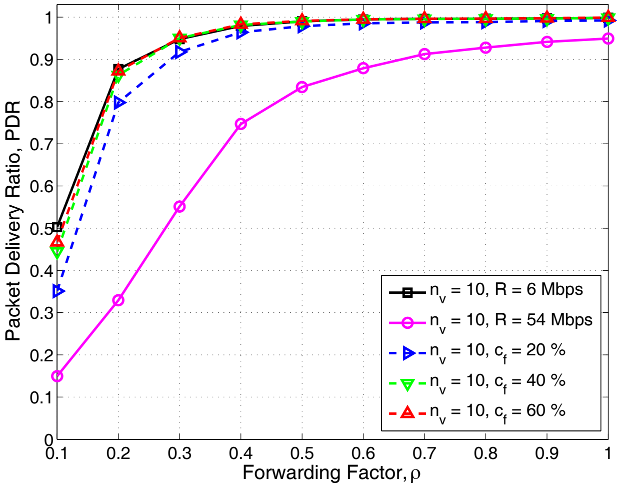

Fig. 10. Packet delivery ratio for a multi-rate ad hoc network as a function of ρ.

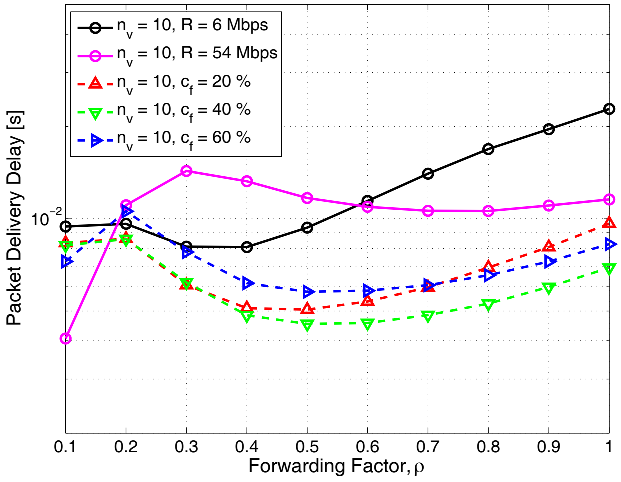

Fig. 11. Packet delivery delay for a multi-rate ad hoc network as a function of ρ.

In this way, each node estimates the status of its neighbors in terms of associated $SINR$. Once node i has this information it updates its internal variables. Subsequently, it considers the fraction c_f of the nodes in its neighboring set that have the highest $\overline{\gamma}$s. c_f is referred to here as *coverage factor*. The data rate at a given node i is thus selected such that all of these nodes will receive packets from node i with a small packet error probability, i.e., smaller than a given threshold P_{th}. In the following results we selected $P_{th} = 0.03$ as it gave good results across all our experiments.

Fig. 10 shows PDR as a function of ρ for different coverage

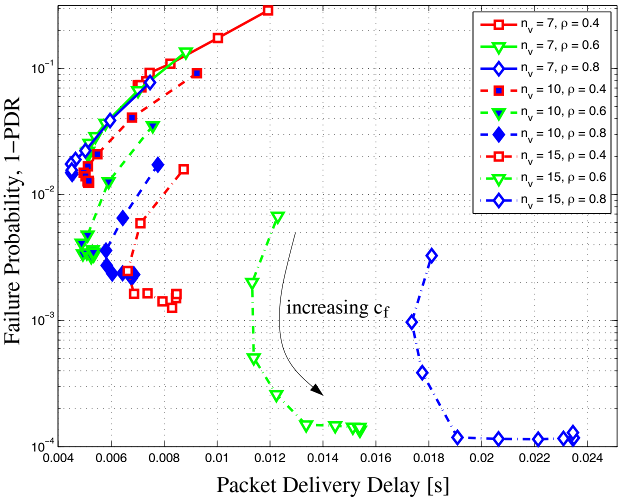

Fig. 12. Tradeoff between failure probability $1 - PDR$ and packet delivery delay for a multi-rate ad hoc network with different network densities. Different curves correspond to different forwarding factors ρ. Each curve is plotted for varying c_f.

factors c_f. For this figure, nodes have on average $n_v = 10$ neighbors, solid curves represent reactive network coding with fixed data rate (only the extreme cases of 6 and 54 Mbps are plotted), whereas dotted curves represent the reliability when nodes are allowed to adapt their data rate based on the above heuristic. First of all, we observe that transmitting with the highest rate of 54 Mbps performs the worst. This is because in such a case the error probability is quite high and this affects the overall performance. As expected, a fixed rate of 6 Mbps leads to the best reliability performance. However, setting $c_f = 40\%$ gets very close to this performance while leading to a better delay metric, as we can see from Fig. 11. From this plot we can further appreciate the benefits of adapting the data rate: overall, for a coverage factor of $c_f = 40\%$ we get the shortest delays by performing, in terms of reliability, almost as well as the fixed rate scheme with 6 Mbps. For the non monotonic behavior of the curves of Fig. 11, note that when ρ is low, e.g., $\rho = 0.1$, the dissemination of innovative information terminates early (in this case deadlocks are frequent) and a large number of nodes are unable to decode all packets (PDR is low, see Fig. 10). An increasing ρ allows the dissemination process to continue longer, leading to fewer deadlocks, thus the delay increases with a corresponding increase in PDR. A further increase of ρ at first allows a quicker dissemination of innovative information (shorter delays), and afterwards the delay increases again due to the increased traffic load (collisions). Fig. 12 shows tradeoff results representing the failure probability $1 - PDR$ as a function of the packet delivery delay for different number of neighbors $n_v \in \{7, 10, 15\}$. Simulation curves are plotted for different values of ρ and each curve is obtained by varying c_f. Good schemes are those lying on the bottom left of the plot (i.e., having short delay and high reliability). From this graph we see that for each (n_v, ρ) pair there exists a suitable coverage factor c_f which minimizes the delay while achieving good performance in term of $1 - PDR$. In practice, c_f in the range $[30, 50]\%$ are good choices as they give good reliability performance while ensuring short delays for all reasonable network sizes. As usual, various tradeoffs can be obtained for different forwarding factors ρ: high ρ values always lead to good PDR performance at the cost of additional delay, while a small ρ is a good choice in terms of delay performance at the cost of an increased failure probability.

V. Conclusions

In this paper we focused on the design of practical broadcasting schemes based on network coding for wireless ad hoc networks. First we discussed the impact of IEEE 802.11-like random access on the performance of reactive network coding [9]. We identified potential problems due to premature termination of the data dissemination. To solve these, we proposed an original *proactive* network coding (*ProNC*) strategy. We finally focused on broadcasting protocols for multi-rate ad hoc networks, where we devised a lightweight rate adaptation heuristic. The designed protocols use estimates of the local node density to reduce the collision probability and keep a sufficient amount of innovative packets flowing across the network. The effectiveness of our schemes was demonstrated by simulation in a range of different scenarios.

Acknowledgment

This paper is partly based on research presented at IEEE ICC 2007 and IEEE GLOBECOM 2007. This work was partially supported by DoCoMo Euro-Labs.

References

[1] P. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulie, “Epidemic information dissemination in distributed systems,” *Computer*, vol. 37, no. 5, pp. 60–67, May 2004.

[2] D. Tuninetti and C. Fragouli, “Processing along the way: forwarding vs. coding,” in *ISITA*, Parma, Italy, Oct. 2004.

[3] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[4] J. Yuan, Z. Li, W. Yu, and B. Li, “A cross-layer optimization framework for multihop multicast in wireless mesh networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2092–2103, Nov. 2006.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 243–254, Oct. 2006.

[6] S. Omiwade, R. Zheng, and C. Hua, “Practical localized network coding in wireless mesh networks,” in *Proc. IEEE SECON*, San Francisco, USA, June 2008, pp. 332–340.

[7] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, “Network planning in wireless ad hoc networks: a cross-layer approach,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 136–150, 2005.

[8] H. Yomo and P. Popovski, “Opportunistic scheduling for wireless network coding,” *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 2766–2770, June 2009.

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

[10] C. Gkantsidis and P. R. Rodriguez, “Network coding for large scale content distribution,” in *Proc. IEEE INFOCOM*, Miami, FL, USA, Mar. 2005, pp. 2235–2245.

[11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[12] R. Koetter and M. Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[13] C. Fragouli and E. Soljanin, “Information flow decomposition for network coding,” *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 829–848, Mar. 2006.

[14] C. Chekuri, C. Fragouli, and E. Soljanin, “On average throughput and alphabet size in network coding,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2410–2424, June 2006.

[15] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *41st Allerton Conference on Communication Control and Computing*, Allerton, IL, USA, Oct. 2003.

[16] H. Dubois-Ferriére, D. Estrin, and M. Vetterli, "Packet combining in sensor networks," in *Proc. ACM SenSys*, San Diego, CA, USA, Nov. 2005, pp. 102–115.

[17] Y. E. Sagduyu and A. Ephremides, "Crosslayer design for distributed MAC and network coding in wireless ad hoc networks," in *Proc. IEEE ISIT*, Adelaide, Australia, Sep. 2005, pp. 1863–1867.

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

[19] "ns2 simulation code for wireless network coding." [Online]. Available: http://telecom.dei.unipd.it/simulation

[20] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "ns2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2," in *Proc. ACM ValueTools*, Nantes, France, Oct. 2007, pp. 1–8.

[21] J. Widmer, C. Fragouli, and J.-Y. L. Boudec, "Low-complexity energy-efficient broadcasting in wireless ad hoc networks using network coding," in *NetCod*, Riva del Garda, Italy, Apr. 2005.

[22] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[23] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin, "Unsaturated throughput analysis of IEEE 802.11 in the presence of non ideal transmission channel and capture effects," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1276–1286, Apr. 2008.

[24] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "On MAC scheduling and packet combination strategies for random network coding," in *Proc. IEEE ICC*, Glasgow, UK, June 2007, pp. 3582–3589.

[25] ——, "A proactive network coding strategy for pervasive wireless networking," in *Proc. IEEE GLOBECOM*, Washington DC, USA, Nov. 2007, pp. 5271–5276.

[26] A. Gkelias, M. Dohler, and H. Aghvami, "Throughput analysis for wireless multi-hop CSMA," in *Proc. IEEE PIMRC*, Barcelona, Spain, Sep. 2004, pp. 979–983.

Alfred Asterjadhi was born in Elbasan, Albania, in 1982. He received the Bachelor's and Master's degree (summa cum laude) in Telecommunications Engineering in 2004 and 2007, respectively, from the University of Padova, Italy. During the Academic Year 2004/2005 he attended several graduate courses of Computer Science at the Rey Juan Carlos University (URJC), Madrid (Spain). Since 2008 he has been a Ph.D. Student at the Department of Information Engineering, University of Padova and is actively participating in national and international research projects. His current research is focused on the design and evaluation of protocols for cognitive wireless networks. Active research is carried out in several sub-fields such as network coding, cross-layer techniques and machine learning.

Elena Fasolo received the Laurea degree in Telecommunication Engineering from the University of Padova in 2004. From January 2005 to December 2007 she was a Ph.D. Student at the School of Information Engineering at the Information Engineering Department, University of Padova, Italy. During this period, she spent six months at DoCoMo Euro-Labs in Munich, Germany, working on network coding techniques for wireless networks. On March 2008 she defended her Ph.D. thesis on network coding strategies for wireless ad-hoc networks. She now works at IKS, Padova, as Application and Network Performance Optimization Specialist.

Michele Rossi received the Laurea degree in Electrical Engineering (with honors) and the Ph.D. degree in Information Engineering from the University of Ferrara in 2000 and 2004, respectively. From March 2000 to October 2005 he has been a Research Fellow at the Department of Engineering of the University of Ferrara. During 2003 he was on leave at the Center for Wireless Communications (CWC) at the University of California San Diego (UCSD), where he performed research on wireless sensor networks. In November 2005 he joined the Department of Information Engineering of the University of Padova, Italy, where he is an Assistant Professor. Dr. Rossi is currently part of the EU-funded SENSEI project and of the WISE-WAI project, both on wireless sensor networks. His research interests are centered around the dissemination of data in distributed ad hoc and wireless sensor networks, including integrated MAC/routing schemes, data dissemination via network coding, the application of compressive sensing techniques for the reconstruction of signals in wireless sensor networks and cooperative routing policies for ad hoc networks.

Joerg Widmer is manager of the Ubiquitous Networking Research Group at DOCOMO Euro-Labs, Munich, Germany. His research expertise covers wired and wireless networks, ranging from MAC layer design for wireless communication, sensor networking, and network coding to transport protocols and future Internet architectures. Before joining DOCOMO, Joerg Widmer was senior researcher at EPFL, Switzerland, working on ultra-wide band communication and network coding. Joerg Widmer obtained his M.S. degree and Ph.D. degree in computer science from the University of Mannheim, Germany in 2000 and 2003, respectively. In 1999 and 2000 he was a visiting researcher at the International Computer Science Institute in Berkeley, CA, USA. He authored more than 60 conference and journal papers, holds several patents, and regularly participates in program committees of major conferences.

Michele Zorzi was born in Venice, Italy, in 1966. He received the Laurea degree and the Ph.D. degree in Electrical Engineering from the University of Padova, Italy, in 1990 and 1994, respectively. During the Academic Year 1992/93, he was on leave at the University of California, San Diego (UCSD), attending graduate courses and doing research on multiple access in mobile radio networks. In 1993, he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, in 1998 he joined the School of Engineering of the University of Ferrara, Italy, and in 2003 joined the Department of Information Engineering of the University of Padova, Italy, where he is currently a Professor. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks, energy constrained communications protocols, and cognitive radio and networks.

Dr. Zorzi was the Editor-In-Chief of the IEEE WIRELESS COMMUNICATIONS MAGAZINE from 2003 to 2005, is currently the Editor-In-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS, and serves on the Steering Committee of the IEEE TRANSACTIONS ON MOBILE COMPUTING, and on the Editorial Boards of the the WILEY JOURNAL OF WIRELESS COMMUNICATIONS AND MOBILE COMPUTING and the ACM/URSI/KLUWER JOURNAL OF WIRELESS NETWORKS. He was also guest editor for special issues in the IEEE PERSONAL COMMUNICATIONS MAGAZINE (Energy Management in Personal Communications Systems) and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Multi-media Network Radios).