Phoenix: Making Cooperation more Efficient through Network Coding in Wireless Networks

Andrea Munari, *Student Member, IEEE*, Francesco Rossetto, *Member, IEEE*,
Michele Zorzi, *Fellow, IEEE*

***Abstract*—This paper introduces a novel MAC protocol for wireless networks, called Phoenix, that employs ideas from Network Coding to enhance decode and forward cooperation. A relay is allowed to code data of its own together with a corrupted packet during a retransmission at no additional cost in bandwidth. Therefore, while in conventional cooperative protocols a node becomes a relay only to assist other terminals, with our proposal a cooperator can also serve its own traffic. We evaluate Phoenix's performance by means of a theoretical model and extensive simulation campaigns. We show that Phoenix is especially beneficial in multihop settings and interesting gains over benchmark protocols can be achieved.**

***Index Terms*—Network coding, cooperation, ARQ, CSMA, ad hoc networks, multihop wireless networks, MAC protocols.**

I. INTRODUCTION

THE importance and interest about cooperative networks in the wireless research community has steadily grown since Laneman's seminal work [1]. In particular, the ability offered by cooperative retransmissions to provide spatial diversity and hence improve network performance has aroused a significant deal of efforts on the design of physical layers that can maximize such potential gains [2], [3]. Somewhat less attention has been devoted to the interaction of cooperation and the upper layers (e.g., MAC and routing) [4]. Starting from these remarks, our work copes with medium access issues and their impact on the network layer. In particular, in many cooperative protocols, relays help other terminals by performing a retransmission on their behalf, but do not pursue a goal of their interest (e.g., delivering their own data). In other words, they serve somebody else's traffic but are not rewarded for that, except for the fact that the whole system may be more efficient. In turn, the underlying concept of our proposal is to allow a relay to use the cooperative retransmission to deliver its own data as well, so that it may both help another terminal and pursue its own interest. Such a special ARQ phase should be carried out with as little additional cost as possible with respect to an ordinary cooperative phase.

In order to implement this idea, it is necessary to somehow combine frames together. Network Coding (NC) is very useful in this context [5], [6], since it enables nodes to code packets together so as to improve the overall network efficiency. However, standard NC can retrieve information only from correctly delivered Protocol Data Units (PDUs), while our scheme inherently has to deal with retransmissions and thus with corrupted packets, requiring a modified form of NC that can leverage incorrect frames. This issue can be overcome by using a special type of physical layer called MIMO_NC [7] and in particular the so called Super MIMO_NC [8]. MIMO_NC is based on parallels between NC and MIMO, in particular the multiple-input, multiple-output nature of both systems. MIMO_NC applies NC to physical layer PDUs and then decodes them by MIMO signal processing, because the packet mixing due to NC resembles the propagation through a wireless MIMO channel. Indeed, the multiple inputs are the PDUs coded together and the multiple outputs are the received frames. Therefore, this approach works also for single antenna nodes (which we assume throughout the paper). MIMO_NC also enables to use corrupted or redundant packets, something standard NC cannot do. In addition, Super MIMO_NC offers the maximum diversity order (equal to 2) for the corrupted frame. Hence the network coded retransmissions both offer spatial diversity and boost throughput. Furthermore, we remark that MIMO_NC's computational complexity is rather low [7], [8], hence the additional processing cost with respect to a standard cooperative protocol is very limited.

The key idea of our proposal is thus to take advantage of network coding in order to make cooperation more efficient in wireless networks. In recent years, there has been a surge of interest in hybrid cooperative-NC systems [7]–[14]. The main goal is to combine the advantages of these two rather different techniques (cooperation creates redundancy to improve the physical layer performance, NC combines packets so as to reduce the required redundancy at the network level). However, the vast majority of the papers in this field focuses on physical layer performance (e.g., bit error rate, diversity order) [7], [9], [10] or information theoretic metrics (like capacity regions) [11]–[13]. Our work (and its forerunners [15], [16]) is one of the few aimed to create a practical medium

---

Manuscript received November 8, 2008; revised April 6, 2009 and July 2, 2009; accepted July 10, 2009. The associate editor coordinating the review of this paper and approving it for publication was D. In Kim.

A. Munari and M. Zorzi are with the Department of Information Engineering, University of Padova, Via G. Gradenigo 6/B, I–35131 Padova (PD), Italy (e-mail: {munari, michele.zorzi}@dei.unipd.it).

F. Rossetto was with the Department of Information Engineering, University of Padova, Via G. Gradenigo 6/B, I–35131 Padova (PD), Italy. He is now with DLR (German Aerospace Center), Institute of Communications and Navigation, 82234 Weßling, Munich, Germany (e-mail: Francesco.Rossetto@dlr.de).

M. Zorzi is also with the University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0407.

This work is supported in part by the U.S. Army Research Office under the Multi-University Research Initiative (MURI) grant no. W911NF-04-1-0224. The material in this paper was presented in part at IEEE SECON 2008, San Francisco (CA, USA), and at IEEE MILCOM 2008, San Diego (CA, USA).

Digital Object Identifier 10.1109/TWC.2009.081478.

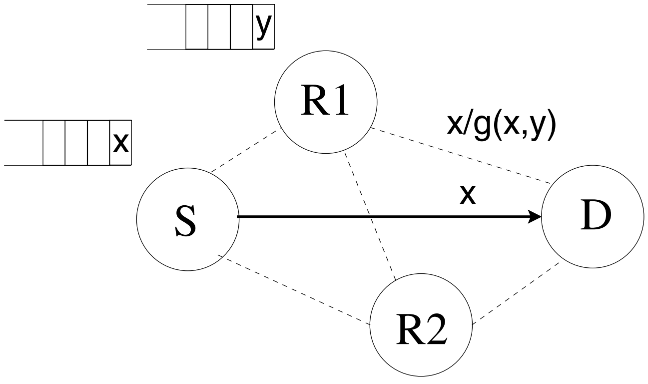

Fig. 1. Reference topology for the protocol description. $g(x, y) = \alpha x + \beta y$ represents a linear combination of packets x and y.

access policy based on a hybrid cooperative-NC physical layer, and the outcome is an IEEE 802.11-style MAC protocol called Phoenix. To the best of our knowledge, the only other research effort that has developed an actual protocol based on these ideas is [14], which implements in a real-world testbed the concept of Analog Network Coding [12]. However, [14] only analyzes toy topologies (composed by up to 5 nodes), while Phoenix has been tested by simulation in a variety of network settings [15], [16]. In addition, while it is not clear how the concept in [14] can scale with the network size or node arrangement, Phoenix can work on systems of arbitrary dimension and has been analyzed in scenarios with up to 100 nodes [16].

The rest of the paper is organized as follows. Section II describes Phoenix and a benchmark cooperative protocol. An analytical model is developed in Section III so as to gain insight on the advantages offered by coded retransmissions. Extensive simulation results in Section IV evaluate the performance of our protocol and finally Section V draws the conclusions.

II. PHOENIX: A HYBRID COOPERATIVE-NC PROTOCOL

In this work we present Phoenix, a novel MAC protocol that leverages cooperative relaying and Network Coding techniques to improve ARQ in Ad Hoc Networks. Phoenix enhances the IEEE 802.11 CSMA policy without channel negotiation [17], which is based on Binary Exponential Backoff (BEB) and carrier sensing. In particular, our protocol introduces a fully distributed procedure to take advantage of cooperative retransmissions, and lets relay nodes exploit such communications to serve their own traffic as well. In the remainder of this section we first present an extended version of IEEE 802.11 CSMA that supports traditional decode and forward cooperation (CCSMA, Section II-A). Then, in Section II-B, we describe the Phoenix MAC.

A. CCSMA: A Cooperative CSMA protocol

Let us consider the topology depicted in Fig. 1, and suppose that node S sends packet x to D employing the IEEE 802.11 CSMA medium access policy (referred to as CSMA throughout this paper). In the event of a successful reception, the destination replies with an ACKnowledgement (ACK) packet and the communication ends. On the contrary, if the transmission fails, no feedback is sent, and the source performs another attempt after a backoff interval. This basic ARQ procedure can be improved by taking advantage of cooperative relaying techniques. In fact, due to the broadcast nature of the wireless medium, some nodes (e.g., R_1 or R_2 in Fig. 1) may have successfully decoded x even if it was not intended for them. In the cooperative paradigm, one of these terminals immediately performs the retransmission on behalf of S (i.e., without additional backoff intervals). In this way, the destination is able to perform Chase Combining on two copies of the same packet that have been received over spatially disjoint and therefore statistically independent channels. This form of spatial diversity greatly improves the decoding probability, potentially reducing the number of required retransmissions. However, some additional coordination among nodes is required for cooperation to be effective, for instance in order to determine when hybrid ARQ is needed and who actually has to perform the retransmission. To this aim, we have designed a protocol, called Cooperative CSMA (CCSMA), that suitably extends plain CSMA. Let us consider again Fig. 1. When CCSMA is employed, any terminal (other than the intended addressee) that successfully decodes packet x (e.g., R_1 or R_2), caches it. At the destination side, two conditions may occur if the reception fails: i) the node is not able to decode the header of the packet; or ii) the header is correctly received but the payload is corrupted. The former case may be induced by harsh channel conditions or by the fact that the node was synchronized to another ongoing transmission. In this situation, since the intended destination did not gather any information, no feedback can be provided and CCSMA resorts to a basic ARQ procedure. On the other hand, if ii) occurs,[1] node D becomes aware of the attempt performed by S and triggers a cooperative phase by caching the corrupted version of x and by transmitting a Not ACKnowledgement (NACK) frame asking for a relayed version of the payload. Terminals that receive the NACK packet and that have a cached version of the payload enter a distributed contention phase based on carrier sensing to elect the relay. Each candidate starts a backoff, whose duration n, in slots, is uniformly drawn in the set $\{0, 1, \ldots, CW_{rel} - 1\}$, where CW_{rel} is the length of the contention window for the relay election. When the countdown has reached the last slot, the node senses the medium and compares the aggregate received power P_{agg} with a reference value $\bar{P}$, possibly equal to the carrier sense threshold. If $P_{agg} \geq \bar{P}$, the relay candidate assumes that another terminal is performing the retransmission and therefore gives up the procedure, going back to its own activity. On the contrary, if the medium is sensed free (i.e., $P_{agg} < \bar{P}$), the node re-encodes x, sends a copy of the packet to the original addressee and then returns to its own activity. When a cooperative transmission takes place, the destination performs Chase Combining and provides a feedback to the source node: if the reception succeeds, an ACK is sent, otherwise, a NACK is transmitted. In the latter case, no further cooperative phase is triggered, and S chooses whether to

---

[1] We assume the packet header to be protected by a stronger FEC than the payload and to have a separate CRC.

perform another attempt resorting to the BEB mechanism. The whole procedure (transmission by the source and potential cooperative phase) is iterated until the packet is correctly decoded at the destination or the Short Retry Limit[2] (SRL) is reached.

Let us point out a few remarks on the distributed relaying scheme that we propose. First of all, in CCSMA cooperation is triggered by the destination node, introducing additional overhead with respect to plain CSMA (i.e., NACK packets). Nevertheless, this approach offers important advantages, as it prevents unnecessary relayed transmissions, thus saving resources and reducing the overall interference. Moreover, the described strategy increases the probability of success for cooperative phases, as the constraint of decoding the NACK packet allows only nodes that have sufficiently good channel conditions to the destination to be relay candidates. Secondly, it is worth noticing that each transmission attempt performed by the source is followed in CCSMA by at most one relaying phase. This choice stems from the observation that a failure of a cooperative retransmission is typically due to harsh interference conditions at the destination. Therefore, other attempts are unlikely to succeed unless performed after a sufficiently long interval, and in such a condition the BEB mechanism turns out to be more effective.

B. The Phoenix Protocol

Cooperative relaying procedures both shorten failure recovery phases and improve their reliability with respect to plain CSMA. Nevertheless, with the basic decode and forward paradigm, a relay node is asked to behave in a selfless way and to hope that other terminals will offer a similar support when needed. In order to work well, such an approach requires every node to pool its own resources. However, in a real network, nodes that are experiencing favorable channel conditions are not encouraged to help their neighbors for two main reasons. First, they would offer part of their bandwidth without serving packets in their own queue. Second, especially in CSMA-BEB based protocols, helping other nodes to deliver their PDUs would reduce their contention window, and therefore would increase the number of contending terminals. In this sense, such a behavior would reduce the cooperators' bandwidth.

Starting from these observations, we propose a novel MAC protocol, called Phoenix, that improves CCSMA by reducing the performance loss that a relay would experience as a result of cooperation. The key idea that underpins Phoenix is to allow a relay to transmit a linear combination of a cooperative packet and of a data unit taken from its own queue instead of simply retransmitting a copy of the former. Not only does such a strategy maintain most of the advantages of classical cooperation in terms of fast failure recovery, but also it magnifies them by reducing the drawbacks of taking part in the cooperative process. In order to unleash the potential of hybrid cooperative-NC ARQ, two ingredients are needed: (i) a physical layer that can handle coded retransmissions, and (ii) a proper medium access policy to achieve coordination during hybrid cooperative phases.

With reference to the condition (i), let us consider a node that has cached a corrupted version of packet x. If such a terminal receives a linear combination of x and another PDU, say y, basic NC techniques would not be able to retrieve any information, since only frames that have been successfully received can be used for joint decoding. This problem can be overcome by means of MIMO_NC [7]. Not only does this PHY have the potential to decode both x and y even if only a corrupted version of the first packet is available, but also it preserves the diversity gain offered by standard decode and forward cooperation.[3] We remark that the computational complexity of MIMO_NC is relatively low, as it employs the low-complexity sphere decoding algorithms [18]. This aspect is further eased in the case under analysis, as MIMO_NC has to solve at most a 2×2 system (2 inputs, the information units to retrieve, and 2 outputs, the received packets). Hence, the additional cost of implementing Phoenix on top of an already existing cooperative protocol is indeed small.

Let us now focus in greater detail on the proposed MAC. Consider again the situation of Fig. 1, and suppose that node D has received the header of packet x without being able to decode the payload. If the quality of the cached frame is extremely poor, the decoding probability for a hybrid Cooperative-NC phase might become too low with respect to pure decode and forward relaying. Therefore, before sending a request for cooperation, D checks the average SINR that characterized the corrupted frame. If this value is below a given threshold Λ_{Th}, the terminal sets the one-bit field *NACK_flag* of the NACK packet to 0. Otherwise, the flag is set to 1. Note that this strategy resembles a rate adaptation policy, since it links the number of data units coded together with the SINRs of the channels. Nodes that receive the NACK frame and that have correctly decoded packet x start the relay election phase of CCSMA. The terminal that wins the contention (i.e., that senses the medium free at the end of the backoff), say R_1, determines which type of retransmission is to be performed. If the *NACK_flag* is set to 0 or the relay has no packets in its queue, the CCSMA cooperative procedure takes place. On the contrary, if the destination allowed a coded transmission and R_1 itself has traffic to serve, a hybrid cooperative-NC phase is initiated, distinguishing two conditions as follows.

A) *The relay has at least one packet y for D in its queue.* In this case, the cooperator generates a linear combination of x and y following the network coding principles and transmits the obtained PDU.[4] At the end of the reception, the destination sends two feedback frames, either ACK or NACK: the first one is addressed to the original source S and regards x, while the second one informs R_1 about the decoding of y.[5]

B) *The relay has no packets for D in its queue, but it*

---

[2] The Short Retry Limit is the maximum number of attempts performed at the MAC layer before dropping a packet [17].

[3] With MIMO_NC, the error probability for a hybrid retransmission (i.e., the receiver jointly decodes x and y) is slightly higher than the one of Chase Combining (i.e., the receiver combines two copies of x), yet the diversity order is preserved [7].

[4] We stress that, with reference to Fig. 1, the encoded packet $\alpha x + \beta y$ sent by the relay in this case is neither a simple bitwise XOR nor a sum of modulated waveforms, but instead it is a linear combination of vectors in a Galois field according to NC principles.

[5] Notice that if y is not successfully decoded, the relay keeps the packet in its queue for later transmission.

has a packet y addressed to another node. In such a condition, the cooperator tries to determine whether a hybrid retransmission is appropriate. Indeed, should the addressee of y, say R_2, have no cached version of x (e.g., because it did not synchronize to the transmission by S), it would not be able to extract the payload y from a linear combination of x and y. In this case, not only would a cooperative-NC phase prevent the relay from successfully serving its traffic, but also it would worsen the performance at D's side with respect to Chase Combining. Therefore, the cooperator transmits a Request To Send (RTS) packet addressed to R_2, containing a field that uniquely identifies x. R_2 replies with a Clear To Send packet (CTS) only if the RTS is decoded and the node has in its cache either a correct copy of x or a corrupted version with SINR above Λ_{Th}.[6] If the CTS is not received, the relay falls back on a pure cooperative transmission, as in CCSMA. On the contrary, if the handshake succeeds, the cooperator sends a linear combination of x and y. This message is followed first by an ACK/NACK by D addressed to S, according to the outcome of the decoding of x. Then, R_2 does the same, informing R_1 of the reception of y. In case of a NACK, node S decides whether to perform another attempt after a suitable backoff interval, while R_1 puts y back in its queue for a later time.

Phoenix gives priority to type A) retransmissions since they generate less overhead and interference than type B).[7] Moreover, we remark that the aim of the RTS/CTS procedure is twofold. First, as discussed, it prevents hybrid retransmissions that would have no chance to succeed. Secondly, in a CSMA-based MAC, the exchanged negotiation messages forestall concurrent communications in the neighborhood of both the cooperator and its destination.[8] The consequent reduction of interference is likely to significantly enhance the success probability for the cooperative phase. These reasons justify the additional handshake overhead.

III. ANALYTICAL MODEL

In order to gain insight on the behavior of Phoenix and the other protocols, it is useful to set up an analytical model to compare them. The scenario is the uplink of a three node network composed by two terminals, called A and B, and an access point. These terminals are sorrounded by an infinite population of similar three-node subnetworks, whose positions follow a two dimensional Poisson-point process with density σ_i. Both A and B have always a packet to transmit. The time is slotted, and during each slot one data packet and the corresponding error-free feedback (ACK/NACK) are sent. At any given slot, each subnetwork transmits with probability p_s, and hence the effective interferers density is $\sigma_i p_s$. It stems that the generated interference follows an α-stable distribution [19], where $\alpha = 2/b$ and b is the path loss exponent. Three different protocols are analyzed, and their names are CSMA, CCSMA and Phoenix, for coherence with the rest of the paper. Terminals are $p(X)$-persistent, i.e., each time a node is given the chance to transmit in a slot, it will do so with probability $p(X)$, where $p(X)$ depends on the number of retransmissions X. In other words, a geometric backoff is assumed. This memoryless backoff simplifies modelling but can still accurately reproduce the behavior of wireless LANs [20]. All the schemes are modelled by a Markov chain, which keeps track of the protocol status (which node will perform the next transmission and the SRL) as well as of the channel conditions. Hence, the state can be represented by four variables $(X, \mathrm{N}, H_A, H_B)$, where X is the number of times the packet has already been transmitted $(0 \leq X \leq \mathrm{SRL} - 1)$, N is the node that will transmit in the next frame (N is hence either A or B) and H_A, H_B represent A's and B's channel SIR, respectively. The channel is subject to correlated Rayleigh fading, and the correlation is modelled according to [21], with carrier frequency of 2.4 GHz and Doppler spread of 40 Hz. CSMA has a SRL of 3, while the cooperative protocols have a SRL of 2. In any state, the chain may perform a transmission with probability $p(X)$. Hence, the average steady state transmission probability p_s can be computed as:

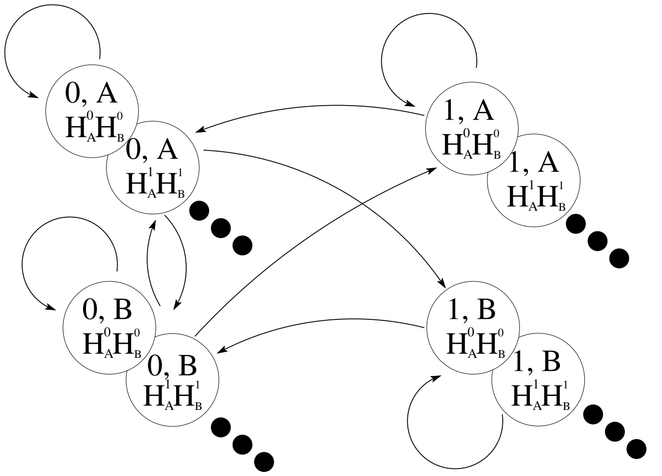

Fig. 2. Representation of the Markov chain for CCSMA and Phoenix

$$p_s = \sum_{\forall \text{ states}} p(X)\pi(X, \mathrm{N}, H_A, H_B) \tag{1}$$

where $\pi(X, \mathrm{N}, H_A, H_B)$ is the steady state probability of being in state $(X, \mathrm{N}, H_A, H_B)$. Of course the $\pi(X, \mathrm{N}, H_A, H_B)$ depend on the packet success probability and hence on p_s. Given an estimate of p_s, the $\pi(X, \mathrm{N}, H_A, H_B)$ are computed and then a new estimate for p_s is evaluated according to (1). The process is iterated until convergence is achieved. One of the virtues of this model is to analyze a simplified interference network based on the above protocols. Fig. 2 shows how the states are organized and outlines the possible transitions for CCSMA and Phoenix.

In CSMA, a node that sends a new packet (say node A), retransmits it until it is correctly delivered or the maximum number of attempts has been reached. When either condi-

---

[6] In Phoenix, unlike CCSMA, a terminal caches the latest received packet whose header has been correctly decoded (be the packet addressed to it or not), regardless of the outcome of the payload decoding.

[7] In type A) retransmissions, since traffic for D is preferred over flows for other nodes, unfairness may arise. However, this effect is confined to strongly asymmetrical topologies.

[8] Notice that terminals in such regions may not be forced to silence by the transmission between S and D.

tion is met, the other node (B in this case) will send its own data. In order to gain a deeper understanding on how the chain works, let us track its evolution in the following situation. Node A has to deliver a new packet for the first time, so the chain starts from state $(0, \mathrm{A}, H_A, H_B)$. Let us call $P_{\mathrm{succ}}(H_A)$ the probability to correctly deliver a PDU given that node A transmits and its channel state is H_A. If A does not get its frame across (which happens with probability $p(0) \cdot (1 - P_{\mathrm{succ}}(H_A))$), the chain transitions into state $(1, \mathrm{A}, H'_A, H'_B)$; if it transmits successfully (probability $p(0) \cdot P_{\mathrm{succ}}(H_A)$) it moves into $(0, \mathrm{B}, H'_A, H'_B)$, otherwise (probability 1-$p(0)$) state $(0, \mathrm{A}, H'_A, H'_B)$ is the destination. If the system is in state $(1, \mathrm{A}, H_A, H_B)$, it may not transmit and transition into $(1, \mathrm{A}, H'_A, H'_B)$ (probability 1-$p(1)$), or it may deliver the packet and move into $(0, \mathrm{B}, H'_A, H'_B)$ (probability $p(1) \cdot P_{\mathrm{succ}}(H_A)$) or finally it may fail and go into state $(2, \mathrm{A}, H'_A, H'_B)$ (probability $p(1) \cdot (1 - P_{\mathrm{succ}}(H_A))$). From state $(2, \mathrm{A}, H_A, H_B)$, either there is a transmission and hence with probability $p(2)$ the chain transitions into $(0, \mathrm{B}, H'_A, H'_B)$ because the SRL has been reached, or the node backs off, going into state $(2, \mathrm{A}, H'_A, H'_B)$.

In CCSMA, if node A fails to deliver its packet to the access point, node B is assumed to have correctly received it and will transmit the payload on behalf of A. The access point will decode A's frame by performing Chase Combining on the two received packets, which convey information on the same data. The transitions for the chain of this protocol have a few subtle but important differences with respect to CSMA. From state $(0, \mathrm{A}, H_A, H_B)$, node A may send unsuccessfully a packet with probability $p(0) \cdot (1 - P_{\mathrm{succ}}(H_A))$ and move into state $(1, \mathrm{B}, H_A, H'_B)$. Note that H_A has not been updated, because the system needs to keep track of the SIR of the first transmitted packet to compute the success probability of the Chase Combining decoding. When node B resends A's frame, the chain transitions into state $(1, \mathrm{B}, H''_A, H''_B)$ and H_A is updated, because B will then transmit a new packet of its own and there is no need to retain the old value of H_A, since it no longer affects the decoding process. However, the chain has not kept memory of how many slots have passed between A's and B's transmissions. Therefore, it cannot exactly compute the transition probabilities as far as A's channel is concerned. Hence the channel evolution is approximated by assuming that the number of elapsed slots is equal the average backoff length, which is $(1 - p(0))/p(0)$.

Finally, Phoenix's chain is identical to CCSMA's, in that the allowed transitions are the same. However, the success probabilities do change, since during a retransmission MIMO_NC has to decode two packets rather than only one. These probabilities have been computed by simulating MIMO_NC decoding. Note that Phoenix here does not employ the threshold mechanism to switch between coded and uncoded retransmissions. All retransmissions embed two packets, because we want to study the impact of MIMO_NC on the system performance and verify whether this switching procedure can be effective or not.

The final goal of our model is to evaluate the throughput per slot per node. This can be computed by associating to each transition a reward, equal to the number of correctly delivered packets [22]. In CSMA and CCSMA, every time a frame is successfully delivered, it yields a reward of one packet. In Phoenix, a successful retransmission yields two data units (the corrupted one plus what the relay coded into the frame).

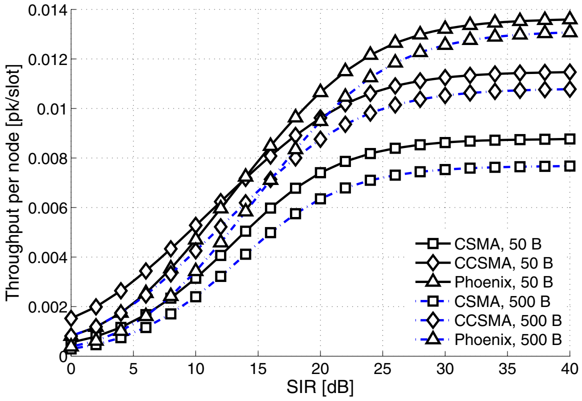

Fig. 3. Throughput per slot per node as a function of the average SIR and the packet size in bytes. The average initial window size is 64 slots

Fig. 3 reports the analytical results on the throughput per slot per node of the described systems as a function of the average SIR. In order to make the scenario homogeneous with those analyzed in the simulation campaign, we have set the value $p(X)$ (one of the model parameters) such that the average backoff lengths in the model and in the simulations are the same. In the latters, the backoff is uniformly chosen in a window 128 slot long, thus the average duration is 64 slots. This implies that in the model $p(X) = 2^{-(6+X)}$, $0 \leq X \leq 2$ for CSMA and $0 \leq X \leq 1$ for CCSMA and Phoenix. Moreover, the results for two packet sizes (500 and 50 bytes) are reported, so as to get insight on the influence of the payload dimension. We point out that the throughput is small in absolute value because it is normalized to the bandwidth, in order to keep the discussion general and not to be bound to a specific data rate.

It is interesting to observe that the magnitudes of the relative gain of Phoenix over CCSMA (about 25 %) and CCSMA over CSMA are roughly the same. This suggests that Phoenix doubles the gains of CCSMA with respect to CSMA. Such a qualitative observation will be confirmed by our simulation results. Also note that a rough upper bound on the throughput is given as follows: an isolated node with errorless channel would transmit on average one packet every $2/CW = 2/128 \simeq 0.0156$ slots, thus the throughput would be roughly 0.0156 pk/slot/node. All protocols are below this bound, but Phoenix is not far from it, confirming the merit of our proposal. Furthermore, the relative gains are rather significant and this is all the more relevant in a CSMA environment, where the potential gains for cooperation are inherently limited by the CS mechanism [23]. It can also be pointed out that there exists a critical SIR value Λ^* such that Phoenix outperforms CCSMA and CSMA for all SIRs larger than Λ^*. This fact confirms the idea that coded retransmissions are useful for higher SIRs and hence supports the mechanism adopted in Phoenix to switch between traditional or NC retransmissions based on the received SINR. In addition, the dependence of Λ^* on the frame size is rather weak. This underpins the choice of a switching threshold Λ_{Th} that does not vary with the PDU length, and shows that the proposed mechanism is suitable for a variety of traffic types.

IV. SIMULATION RESULTS

To evaluate the benefits offered by hybrid cooperative-NC ARQ in more complex scenarios, we have tested the performance of Phoenix by means of extensive simulations using the Omnet++ modeler [24]. Our scheme has been compared to two benchmarks, namely plain CSMA and CCSMA. All the protocols have been studied in a wireless environment subject to correlated Rayleigh fading, with Doppler frequency equal to 40 Hz, corresponding to a speed of 5 m/s at 2.4 GHz. The values of the Short Retry Limit (SRL) have been chosen so that all the MAC schemes offer a similar reliability for single hop flows (see Tab. II). In particular, the SRL for plain CSMA has been set to 3, while 2 independent transmission attempts are sufficient for CCSMA and Phoenix, thanks to the advantages offered by cooperative relaying techniques. As far as the relay election is concerned, the maximum length of the contention window, CW_{rel}, has been identified considering two opposite trends. On the one hand, the larger the CW_{rel} value, the lower the collision probability among cooperators. On the other hand, a lengthy backoff interval reduces the gains of cooperative relaying because of the longer failure recoveries. Starting from these remarks, we have performed some preliminary simulations, and we have determined that a reasonable tradeoff between the two factors for the topologies under study was represented by $CW_{rel} = 32$. Finally, in our simulations we allow a hybrid cooperative-NC phase only if the node asking for a retransmission has in its cache the corrupted frame with SINR $\geq \Lambda_{Th} = 3\,\text{dB}$. This value stems from the BER/SINR tables for MIMO_NC, as with SINRs lower than 3 dB for the cached packet, the decoding probability of a coded retransmission falls below 2/3. A complete list of the parameters used in our simulation campaigns can be found in Tab. I.

We have tested our protocols in both single-hop and multihop networks. In the former scenario, 35 nodes are spread over a $260 \times 260\, m^2$ area, and each of them generates packets addressed to its neighbors according to a Poisson traffic model with intensity λ. This configuration allows multiple simultaneous communications in the network, and tests the protocols when hidden terminals and external interference are present. Such a setting is meaningful to highlight the performance of MAC schemes in harsh medium contention conditions.

Moreover, we have studied Phoenix and its competitors in a multihop environment. In this case, 25 nodes are deployed in a $200 \times 200\, m^2$ square. The considered topologies are connected, and all-to-all Poisson traffic is generated. We assume the routing tables to be known a priori at each terminal. To stress the impact of multihop flows, we have implemented a Random Early Detection (RED) policy [25]. We divide packets processed at terminal A in three classes: $\mathcal{C}_1$ identifies traffic that is generated at the current node; $\mathcal{C}_2$ contains frames that have undergone one or two hops to reach A; while frames that have travelled more than two hops belong to $\mathcal{C}_3$. Furthermore, letting L be the size of the buffer at the MAC layer, we associate a threshold value $T_i = \alpha_i \cdot L$ to each class. When a packet of class $\mathcal{C}_i$ is received from the network layer, our MAC schemes check the number n of frames currently in the buffer. If $n \leq T_i$, the packet is inserted in the queue for later transmission. Otherwise, the packet is accepted with probability $1 - p_d$ and discarded with probability p_d, where $p_d = (n - T_i)/(L - T_i)$. Clearly, the larger α_i (and hence T_i), the smaller the rejection probability p_d. In our studies, $\alpha_1 = 1/2$, $\alpha_2 = 3/4$ and $\alpha_3 = 1$.

TABLE I
PARAMETERS USED IN OUR SIMULATIONS

Transmission power	10 dBm
Noise Floor	-102 dBm
CS threshold	-100 dBm
CS threshold for relay contention, P	-100 dBm
Detection threshold	-96 dBm
Path loss exponent, b	3.5
Maximum Doppler shift	40 Hz (5 m/s)
Slot, DIFS, SIFS duration	20, 128, 28 μs
Carrier Frequency	2.4 GHz
Data Rate B	6 Mbit/s
MAC buffer size, L	24
Initial maximum contention window	128 slots
Short Retry Limit - CCSMA and Phoenix	2
Short Retry Limit - CSMA	3
Number of slots used for relay contention, CW_{rel}	32
Minimum SINR to trigger a cooperative-NC phase, Λ_{Th}	3 dB
Simulation Time	12 s
Simulation Transient (metrics not collected)	3.5 s
DATA header CSMA - CCSMA	272 bits
DATA header Phoenix	280 bits
Payload	2000 bits
ACK/NACK/CTS	112 bits
RTS	160 bits

All the results reported in this section have been averaged over multiple simulations, so that the 95% confidence interval never exceeds 3% of the estimated value. The duration of each simulation was chosen long enough to stabilize the results.

A. Single Hop Networks

The first metric that we consider is the aggregate network throughput, depicted in Fig. 4 against the nominal load per node λ, expressed in kbps. Cooperative relaying techniques show their beneficial effect already at relatively low loads. Their faster and more reliable failure recovery procedures boost the performance of both CCSMA and Phoenix with respect to plain CSMA, with at least a 10% gain at sufficiently high loads. On the other hand, the impact of hybrid cooperative-NC phases becomes more evident as traffic increases, since the more the enqueued packets, the higher the number of coded retransmissions. The capability to serve additional traffic during relaying phases turns out to be profitable in terms of aggregate throughput: at saturation Phoenix outperforms CSMA by 18% and CCSMA by almost 10%. Fig. 4 can also be analyzed in light of the results of Section III. Our analytical model predicted similar throughput percentage gains for Phoenix over CCSMA and for CCSMA over plain CSMA. Such trend is clearly confirmed by the curves of Fig. 4. Incidentally, we notice that the gap between the schemes as computed in Section III is higher than the one achieved in our simulation campaigns. This difference stems from

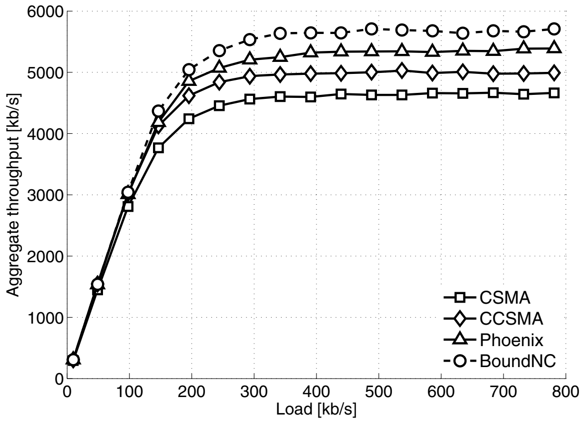

Fig. 4. Aggregate throughput vs. nominal load, single hop network. Payload length 2000 bits, SRL$_{CSMA}$ = 3, SRL$_{CCSMA}$ = 2, SRL$_{Phoenix}$ = 2.

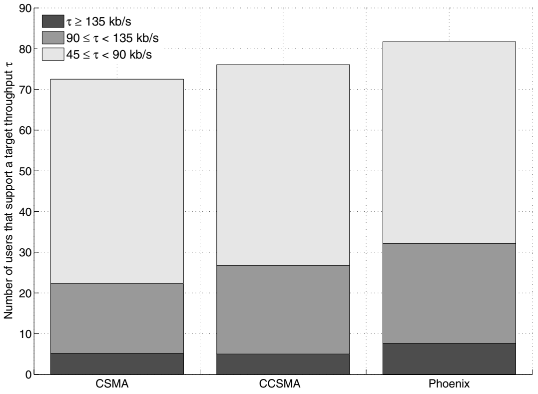

Fig. 5. Number of users that enjoy a given QoS, single hop network. Payload length 2000 bits, SRL$_{CSMA}$ = 3, SRL$_{CCSMA}$ = 2, SRL$_{Phoenix}$ = 2.

medium access issues such as header losses and cooperator unavailability that could not be considered in the analytical model due to complexity but that significantly affect the performance of the protocols in a CSMA environment [23]. To investigate the effectiveness of Phoenix's approach to hybrid cooperative-NC phases, we have also tested a reference protocol, named BoundNC, that implements the proposed medium access policy taking advantage of some ideal assumptions. First, all the cooperative retransmissions performed using BoundNC succeed irrespective of the SINR (i.e., x is decoded in case of Chase Combining or both x and y are retrieved in case of hybrid retransmission). Thus, the protocol allows each cooperative phase to be network-coded by always setting the *NACK_flag* to 1. Secondly, a relay candidate is assumed to be aware of which neighbors have cached a copy of x. Exploiting this information, cooperative-NC retransmissions involving secondary destinations can take place without the RTS-CTS handshake procedure implemented in Phoenix. In view of these two properties, the considered scheme, while obviously impossible to implement in practice, represents an upper bound for the class of protocols that implement hybrid cooperative-NC ARQ relying on a distributed CSMA-based contention for the choice of the relay. We report the results for BoundNC only for the aggregate throughput, since the other metrics would report similar trends. Fig. 4 shows that BoundNC (dashed line) only offers a limited improvement with respect to Phoenix, approximately 5% at saturation. This highlights how our solution indeed represents a good realization of hybrid cooperative-NC ARQ techniques in a CSMA environment, since it achieves most of the available gains at low cost in terms of protocol complexity.

While Fig. 4 dealt with the overall data rate the whole network can support, it is insightful to understand how the throughput is distributed among the nodes. In order to evaluate this aspect, we introduce a reference throughput $\bar{\tau}$ and we divide the users in classes with respect to this target. Let n_c be the number of terminals that are inhibited due to the carrier sense mechanism by the transmission of a given node. It stems that these users cannot serve their traffic during the current communication, and they must defer their access to the channel. In such a system, we define $\bar{\tau}$ as the saturation nominal load per node $\bar{\lambda}$ multiplied by the payload length L. In turn, $\bar{\lambda}$ is the minimum value of λ that fully utilizes the data rate B, i.e., $\bar{\lambda}\, n_c L_e \triangleq B$, where L_e is the shortest time required to perform a complete data transmission multiplied by the data rate. With the parameters used in our simulations, $\bar{\tau} = 150\,\text{kb/s}$. This model does not take into account imperfections of the MAC protocol, like collisions. Therefore, in a real implementation, nodes will achieve only a fraction of $\bar{\tau}$. Fig. 5 depicts the number of users that achieve a certain share S_τ of the target throughput, namely $S_\tau = 90, 60, 30\%$. For example, the darkest bars represent the number of nodes that enjoy at least 90% of $\bar{\tau}$. The improvement offered by Phoenix over its competitors is twofold. On the one hand, the share of terminals that support the minimum reference throughput increases by 8% with respect to CCSMA and by 14% with respect to plain CSMA. On the other hand, our protocol boosts the number of nodes with medium and high QoS by as much as 21% and 45% respectively. These results show that the combination of cooperation and NC can guarantee a minimum service level to a larger population. Let us focus on the trends for CCSMA and CSMA. Two remarks can be made: i) CCSMA increases the number of nodes that support the minimum throughput; ii) the cardinality of the highest QoS class for CCSMA is slightly smaller than the one for CSMA. This offers an interesting insight on the impact of cooperation. With a decode and forward approach, relay nodes spend some of their resources in order to help other terminals. Hence, not only do cooperators reduce their performance, but also terminals that benefit from this help become more aggressive, as their higher success rate leads them to contend for the channel more often. Both these factors are detrimental for relays. We can then infer that cooperation redistributes the resources in the network at the expense of users with high QoS. This effect, on the contrary, has no impact on Phoenix: MIMO_NC does not disadvantage relays, since they can deliver their own traffic as well.

Fig. 6 shows the average transmission energy consumption per successfully acknowledged information bit against the nominal load. First of all, we notice that the CSMA curve

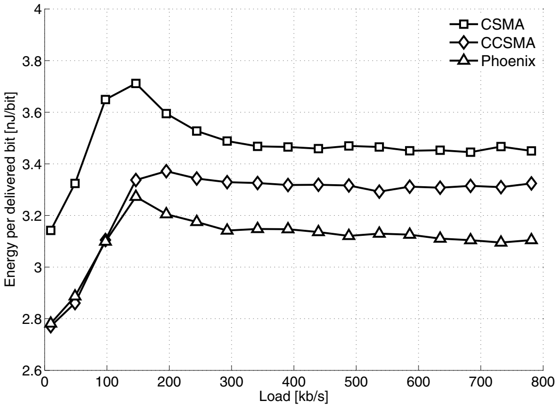

Fig. 6. Transmission energy consumption per delivered bit vs. nominal load, single hop network. Payload length 2000 bits, SRL_{CSMA} = 3, SRL_{CCSMA} = 2, $\text{SRL}_{Phoenix}$ = 2.

decreases as the traffic increases beyond a certain point. This effect is due to the unfairness that characterizes medium access schemes based on carrier sensing. With such protocols, nodes that experience poor average link conditions, e.g., because they are far from their destination, incur longer backoff cycles and therefore tend to access the medium with lower frequency. On the contrary, terminals that enjoy favorable positions are inclined to transmit with shorter contention phases and with higher success probability, grabbing a larger share of the bandwidth. Such a behavior becomes more evident as traffic increases. Therefore, at high nominal loads most of the transmissions are performed by nodes that belong to the latter class, resulting in a lower average cost in terms of energy. The impact of unfairness in CSMA leads to an almost 10% drop in energy consumption from low to high loads. On the other hand, this effect is far less pronounced for CCSMA. This stems from the beneficial influence of cooperation, that shortens ARQ phases also for terminals that experience bad channel conditions, preventing them from being stuck in backoff cycles and thus favoring a more fair distribution of the resources in the network. As far as Phoenix is concerned, a drop in energy consumption at high loads can be noted. However, unlike for CCSMA, this trend is not due to unfairness, but to the higher number of coded retransmissions that take place in such conditions, which make it possible to deliver packets at no additional cost in terms of energy. Two further observations can be made on Fig. 6. First, at low traffic rates, the cooperative protocols outperform CSMA by as much as 20%. In these conditions, all the terminals manage to access the medium, and the higher number of retransmissions required to successfully deliver a packet, due both to the lack of spatial diversity and to the higher SRL, highlights the energy inefficiency of plain CSMA. Secondly, at high loads Phoenix provides some improvement over CCSMA, which stems from the capability of our protocol to deliver additional information bits during cooperative phases at no cost in terms of energy and bandwidth.

In our work, we have also studied the dependence of the different schemes on some protocol parameters, namely SRL and payload length. The results of these analyses are reported at saturation in Tab. II. Let us first focus on the impact of the former parameter. As expected, higher values of SRL raise the reliability, i.e., the Packet Delivery Ratio (PDR), and the average latency, due to the increased number of retransmission attempts. Incidentally, we notice that Phoenix slightly improves the PDR over CCSMA for all values of the SRL. This effect again stems from the high reliability of cooperative-NC phases induced by a suitable choice of Λ_{Th}. Moreover, by studying Jain's index, not reported here due to space constraints, we have noticed that fairness decreases as the value of SRL increases. This can be explained observing that the larger the number of retransmissions, the longer the backoff cycles that nodes are likely to enter, and therefore the less homogeneous the bandwidth distribution in the network. The impact of unfairness can be seen in the decreasing trends for energy consumption, outage delay and outage throughput.[9] We remark that Phoenix outperforms its competitors for all the metrics in all the considered scenarios, but these advantages mildly shrink as the SRL increases. This is an effect of the longer backoffs that nodes experience for higher SRL, which improve the temporal diversity at the receiver. As a consequence, the impact of spatial diversity is reduced. Incidentally, we stress that Tab. II confirms how the reference values of SRL used for CSMA and for the cooperative protocols in our simulations have been chosen in order to provide a comparable PDR.

Let us now consider the performance dependence of the MAC schemes on the payload length. As is reasonable to expect, some metrics (like PDR or outage delay) worsen as the packet length increases. In addition, it is important to notice that the gains of Phoenix over the other protocols are roughly invariant with respect to the frame size. Therefore, our scheme works well for different packet dimensions, as the analytical model of Section III predicted.

B. Multihop Networks

In Section IV-A we discussed the performance of Phoenix and its competitors in a scenario apt to stress harsh channel contention. To complete our analysis, we now focus on the behavior of the protocols in multihop networks. In this case, data packets may not be correctly delivered because of PHY related impairments (e.g., fading and interference) or MAC level problems like buffer overflow. It is well known that the latter issue tends to reduce the share of successfully delivered multihop traffic, in favor of single hop. However, the aim of our analysis is to specifically evaluate the advantages brought by hybrid cooperative-NC retransmissions on longer routes. Therefore, we have implemented RED in all the protocols, whose objective is to reduce the impact of losses due to a full buffer especially for multihop flows. The beneficial effect of RED is apparent in Fig. 7, where the share of multihop traffic increases for Phoenix from 16% to 30% at high loads.

Fig. 8 presents the PDR achieved by the considered MAC policies for different route lengths, depicted against the nominal load. Single hop traffic enjoys a high reliability with all

---

[9] The outage throughput (delay) is defined here as the 20-th (80-th) percentile of the throughput (delay) experienced by all the nodes in the network.

TABLE II
PARAMETRIC STUDIES ON SRL (WITH PAYLOAD SET TO 2000 BITS) AND PAYLOAD LENGTH (WITH $\mathrm{SRL}_{CSMA} = 3$, $\mathrm{SRL}_{CCSMA} = 2$, $\mathrm{SRL}_{Phoenix} = 2$) FOR THE SINGLE HOP SCENARIO. RESULTS ARE REPORTED AT SATURATION LOAD.

Protocol	Parameter	PDR (%)	Delay [ms]	Energy [nJ/bit]	Out. Del. [ms]	Out. Thr. [kb/s]
CSMA	SRL 3	85.0	289	3.45	505	72.2
	SRL 4	90.9	323	3.30	597	65.7
	SRL 5	95.0	348	3.09	730	58.0
	SRL 6	97.5	369	2.90	818	52.6
CCSMA	SRL 2	85.5	277	3.32	458	80.1
	SRL 3	93.6	316	3.27	558	75.5
	SRL 4	98.1	333	3.12	616	73.7
Phoenix	SRL 2	86.1	260	3.10	416	91.8
	SRL 3	93.6	298	3.07	531	79.9
	SRL 4	97.5	322	2.99	601	77.2
CSMA	1000 bits	88.7	211	1.79	337	112
	2000 bits	85.0	289	3.45	505	72.2
	2500 bits	83.5	328	4.37	570	60.6
CCSMA	1000 bits	89.7	204	1.75	312	128
	2000 bits	85.5	277	3.32	458	80.1
	2500 bits	83.5	311	4.14	520	69.2
Phoenix	1000 bits	90.1	193	1.65	280	148
	2000 bits	86.1	260	3.10	416	91.8
	2500 bits	84.8	296	3.88	492	72.9

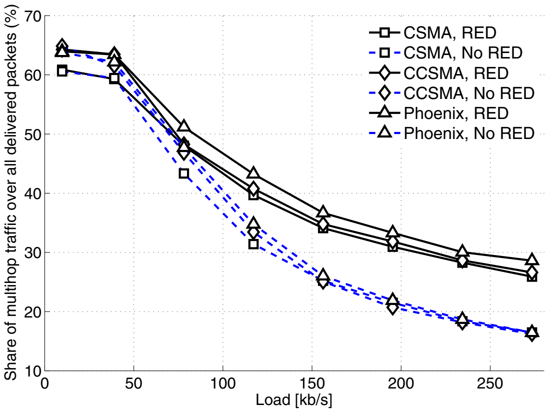

Fig. 7. Comparison of the performance of the protocols with and without Random Early Depletion (RED): share of multihop traffic successfully delivered over total delivered traffic.

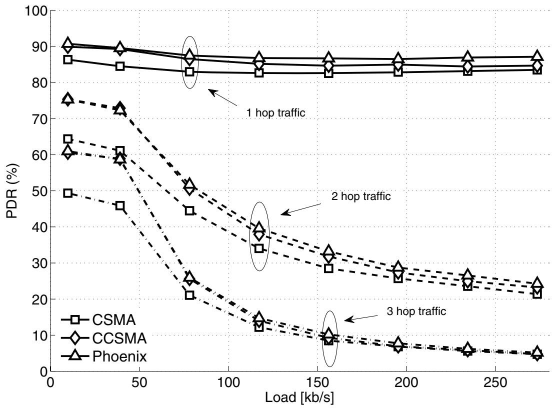

Fig. 8. Average Packet Delivery Ratio as a function of nominal load. Each set of curves shows the behavior for a specific route length in hops.

the protocols, whereas multihop paths, as expected, incur more losses. The impact of cooperation is apparent: CCSMA and Phoenix improve the PDR with respect to CSMA by 10% for two hop routes and by up to 15% for paths composed by three hops. The benefit stems from the better failure recovery capabilities of relaying techniques with respect to plain ARQ, that help to keep alive multihop flows. This effect is even magnified by hybrid cooperative-NC procedures, as shown by the fact that Phoenix outperforms CCSMA regardless of the load. The ability to exploit relaying phases to serve additional traffic has two main consequences: not only do coded packets proceed closer to their destination, but also the saturation of MAC queues is slowed down, positively affecting the whole network. Moreover, the high reliability offered to multihop traffic is particularly beneficial to the overall system, since it repays the network for the efforts made to route packets through several hops.

Another metric of interest is the aggregate throughput, reported in Fig. 9 according to the distance in hops. First of all, we notice that the enhancements offered by Phoenix for single hop flows, almost 10% over CCSMA, confirm the trends discussed in Section IV-A. On the other hand, the gains of our protocol are magnified when multihop paths are considered: for two hop routes Phoenix beats CCSMA by more than 10% and plain CSMA by more than 25%, while in three hop paths the improvements are up to 18% and almost 30% respectively. These boosts stem once again from the higher fairness that Phoenix provides to frames that travel longer distances.

In conclusion, we consider the average end-to-end delay, depicted against the nominal load in Fig. 10. The plot clearly shows two main trends. On the one hand, cooperative relaying helps in containing the latency with respect to plain CSMA thanks to the shorter failure recovery procedures. On the other hand, coded retransmissions make it possible to both reduce the time packets have to spend in the buffer and to avoid delays due to medium access contention. The impact of these beneficial effects is proportional to the number of hops a frame has to undergo to reach its final destination, and the performance advantages offered by Phoenix become

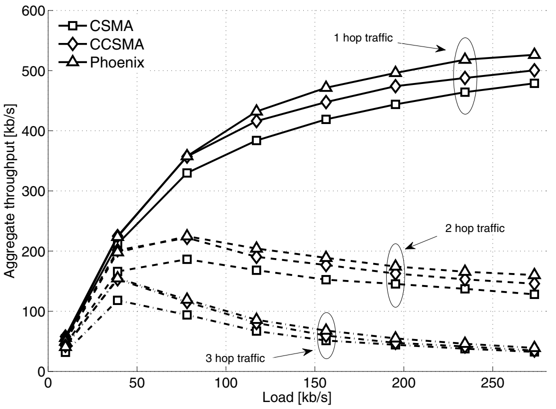

Fig. 9. Aggregate throughput as a function of nominal load. Each set of curves shows the behavior for a specific route length in hops.

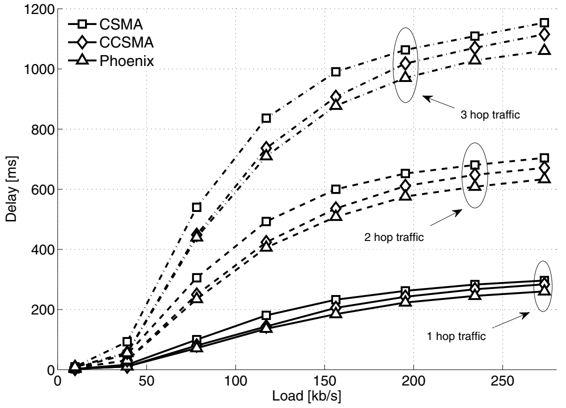

Fig. 10. Average end to end delay as a function of nominal load. Each set of curves shows the behavior for a specific route length in hops.

evident especially for three hop routes, where our protocol reduces the average latency by as much as 40 ms (16%) with respect to CSMA and by 25 ms (10%) compared to plain decode and forward cooperation. We infer that not only does Phoenix assure a larger number of multihop communications (see Fig. 9) with higher reliability, but also it is able to deliver the payload much more quickly.

V. CONCLUSIONS AND FINAL REMARKS

In this paper, we have proposed a new cooperative protocol (Phoenix) that hinges on Network Coding (NC) to achieve improved performance. In particular, NC is used to allow nodes to deliver their own data frames during a retransmission, hence providing another good reason to cooperate. In contrast to most of the existing literature, which focuses on physical layer metrics, we devoted our attention to network and protocol design. Phoenix has been tested in a variety of environments [15], [16] and the following lessons and conclusions have been drawn:

- In single hop networks, Phoenix yields the best benefits for delay constrained applications, i.e., for low SRL. In addition, the possibility to cooperate and pursue their own interest does not reduce the cooperators' performance;
- In clustered networks [16], gains over CCSMA are somewhat larger than for random single hop networks (around 12%) and they scale with the network size. Relays are often nodes close to the gateway, and their throughput is not lowered by cooperation. Instead, the number of users with high QoS is significantly expanded with respect to both CSMA and CCSMA. Also coverage is improved because of the larger number of low QoS nodes;
- In tree networks [15], Phoenix is particularly useful, since such topologies have plenty of bottlenecks, especially close to the sink, and network coded cooperation relieves congestion on them and avoids that packet losses may delay traffic of all upstream nodes. It follows that more fairness is achievable;
- In mesh networks, gains improve with the route length, because the reduction of queueing times is especially beneficial to multihop traffic;
- The protocol is relatively simple compared to CCSMA and also to CSMA. The only remarkable difference with respect to CCSMA lies in the type of cooperation (which is more a PHY problem rather than a MAC issue, and entails only a slight increase in the computational complexity), while both CCSMA and Phoenix must implement a distributed relay election phase. This procedure is heavily based on the IEEE 802.11 backoff mechanism and hence is a rather straightforward software upgrade.

In conclusion, the introduction of network coded cooperation has proven to yield important gains also in more realistic network scenarios than the simple topologies tested in the past and Phoenix has been shown to perform well in a variety of environments.

ACKNOWLEDGMENT

The authors would like to thank Elena Fasolo for her work on Phoenix. They would also like to thank the associate editor and the anonymous reviewers for their insightful comments.

REFERENCES

[1] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3518-3539, Dec. 2004.

[2] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 68-732, Oct. 2004.

[3] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3037-3063, Sept. 2005.

[4] F. Librino, M. Levorato, and M. Zorzi, "Distributed cooperative routing and hybrid ARQ in MIMO-BLAST ad hoc networks," in *IEEE GLOBECOM*, Washington (DC, USA), Nov. 2007.

[5] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204-1216, July 2000.

[6] P. A. Chou, T. Wu, and K. Jain, "Practical network coding," in *41st Allerton Conf. Commun., Control Computing*, Monticello (IL, USA), Oct. 2003.

[7] E. Fasolo, F. Rossetto, and M. Zorzi, "Network coding meets MIMO," in *NetCod 2008*, Hong Kong (China), Jan. 3-4 2008.

[8] ——, "On encoding and rate adaptation for MIMO_NC," in *IEEE 3rd International Symp. Commun., Control Signal Processing 2008*, Malta, 12-14 Mar. 2008.

[9] L. Xiao, T. E. Fuja, J. Kliewer, and D. J. Costello Jr., "A network coding approach to cooperative diversity," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3714-3722, Oct. 2007.

[10] X. Bao and J. Li, “Adaptive network coded cooperation (ANCC) for wireless relay networks: Matching code-on-graph with network-on-graph," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 574-583, Feb. 2008.

[11] B. Nazer and M. Gastpar, “Computation over multiple access channels," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3498-3516, Oct. 2007.

[12] S. Katti, I. Maric, A. Goldsmith, D. Katabi, and M. Medard, “Joint relaying and network coding in wireless networks," in *IEEE ISIT 07*, Nice (France), 24-28 June 2007.

[13] C. Peng, Q. Zhang, M. Zhao, and Y. Yao, “On the performance analysis of network-coded cooperation in wireless networks," in *IEEE INFOCOM 2007*, Anchorage (AK, USA), 6-12 May 2007.

[14] S. Katti, S. Gollakota, and D. Katabi, “Embracing wireless interference: Analog network coding," in *ACM SIGCOMM 07*, Kyoto (Japan), 27-31 Aug. 2007.

[15] E. Fasolo, A. Munari, F. Rossetto, and M. Zorzi, “Phoenix: A hybrid cooperative-network coding protocol for fast failure recovery in ad hoc networks," in *IEEE SECON 2008*, San Francisco (CA, USA), 16-20 June 2008.

[16] A. Munari, F. Rossetto, and M. Zorzi, “On the viability of a cooperative-network coding protocol in clustered networks," in *IEEE MILCOM*, San Diego (CA, USA), Nov. 17-19 2008.

[17] IEEE LAN MAN Standards, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications high-speed physical layer in the 5 GHz band," ANSI/IEEE Std., Sept. 1999.

[18] H. Vikalo and B. Hassibi, “On joint detection and decoding of linear block codes on Gaussian vector channels," *IEEE Trans. Signal Processing*, vol. 54, no. 9, pp. 3330-3342, Sept. 2006.

[19] E. S. Sousa and J. A. Silvester, “Optimum transmission ranges in direct sequence spread spectrum multihop packet radio network," *IEEE J. Select. Areas Commun.*, vol. 8, no. 5, pp. 762-770, June 1990.

[20] F. Calì, M. Conti, and E. Gregori, “Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Networking*, vol. 8, no. 6, pp. 785-799, Dec. 2000.

[21] H. S. Wang and N. Moayeri, “Finite state Markov channel - a useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163-171, Feb. 1995.

[22] M. Zorzi and R. R. Rao, “Error control and energy consumption in communications for nomadic computing," *IEEE Trans. Comput.*, vol. 46, no. 3, pp. 279-289, Mar. 1997.

[23] M. Levorato, A. Munari, and M. Zorzi, “On the effectiveness of cooperation in carrier sense-based ad hoc networks," in *IEEE SECON*, Rome (Italy), June 2009.

[24] A. Varga, “Omnet++," [Online]. Available: http://www.omnetpp.org, 2001.

[25] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397-413, Aug. 1993.

Andrea Munari (S’06) was born in Venice on April 25th, 1981. He received the Laurea degree (MSc) in Telecommunications Engineering *summa cum laude* from the University of Padova, Italy, in 2006. In January 2007 he joined the Department of Information Engineering at the same university, where he is currently a Ph.D. student under the supervision of Prof. Michele Zorzi. Since 2007 he has also been collaborating with IBM Zurich Research Laboratory, Switzerland, focusing on the design of routing protocols for wireless sensor networks with particular attention to energy efficiency. His research interests include cooperative techniques and directional communications in wireless ad hoc networks.

Francesco Rossetto (S’06, M’09) received the "laurea" (equivalent to MS) and the PhD in Telecommunications Engineering in 2005 and 2009, respectively, from the University of Padova, Padova, Italy. In 2004-2005 he studied electrical engineering at the University of California, San Diego, CA, USA, under a student exchange program. In 2008 he was on leave at the University of California, San Diego, working for the MURI project, a multiuniversity initiative for the development of multihop MIMO networks. Since 2009 he is with the DLR (German Aerospace Center) in Munich, Germany. His research interests include satellite communication, network coding and cross layer design. His corporate experience includes a summer internship in 2006 at the Ericsson Eurolabs, in Aachen, Germany, working on Hybrid ARQ for 3G/LTE cellular networks.

Michele Zorzi was born in Venice, Italy, on December 6th, 1966. He received the Laurea Degree and the Ph.D. 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, where he became a Professor in 2000. Since November 2003, he has been on the faculty at the Information Engineering Department of the University of Padova. 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 broadband wireless access.

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 IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the WILEY JOURNAL OF WIRELESS COMMUNICATIONSAND 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).