On the viability of a Cooperative-Network Coding Protocol in Clustered Networks

Andrea Munari*, Francesco Rossetto*, Michele Zorzi†

*DEI - University of Padova, Via G. Gradenigo, 6/B, Padova, 35131, Italy
†California Institute for Telecommunications and Information Technology – UC San Diego, USA
{munari, rossettm, zorzi}@dei.unipd.it

Abstract—The interaction between Cooperation and Network Coding has lately received a significant deal of attention, as a combination of the two brings novelty, flexibility and improved performance. However, there is a lack of studies about real-world scenarios. In this paper, we analyze the performance of an existing Cooperative-NC protocol for clustered networks, where all nodes connect in a single hop fashion to a cluster head. This is representative of environments like WLANs, mesh networks, last mile connectivity or ad hoc networks where some nodes create an infrastructure for the others. We show that our system is able to improve network throughput by as much as 15% and network capacity by 20% over decode-and-forward cooperative protocols.

I. INTRODUCTION AND RELATED WORK

Cooperation [1] and Network Coding [2], [3] are two powerful techniques that have enjoyed a high degree of interest during this decade. The former improves the reliability of wireless communications by having different nodes transmit the same information. This redundancy comes from distinct terminals and thus leads to spatial diversity. Hence, cooperation can reduce the error rate very quickly as the average SNR increases. However, this approach requires the relay nodes to use their own resources (bandwidth and energy) to help other terminals while receiving no immediate benefit (e.g., delivering their own traffic). On the other hand, Network Coding (NC) is a throughput efficient strategy that enables nodes to code packets together rather than just transmitting them separately. Extensive studies have shown that significant improvements are attainable for both throughput and delay, e.g. [3], [4]. However, since this efficiency stems from a clever reduction of redundancy, NC is particularly vulnerable to high error rates and this can be a substantial handicap in wireless environments.

Thus these two techniques have opposite weaknesses, and in the past two years several attempts have tried to combine the best of both of them [5]–[9]. However, most of these studies analyze physical layer or information theory metrics, such as diversity order, capacity regions or outage probabilities. In all cases, very simple MAC protocols are considered and some centralized scheduler is often assumed, so as to study the aforementioned metrics without undesired cross-layer interactions. Moreover, the protocols are tested on toy topologies, which again avoid coupling between PHY and higher layers but leave open the question of the actual performance in more complex, realistic networks with large, random node deployments.

A remarkably different approach has been undertaken in [10], where a special type of enhanced NC (named MIMO_NC1 [9]) is used during ARQ retransmissions. The proposed protocol (called Phoenix) makes significant efforts towards practical implementability, since the only notable assumption is the adoption of MIMO_NC as part of the PHY. In addition, [10] is focused on network level rather than physical layer performance and analyzes systems composed by up to 50 randomly located nodes.

The main goal of this paper is to assess the performance of Phoenix in clustered networks and to gain more insight into the behavior of this protocol. By clustered network we mean a wireless system where nodes have single hop connectivity to a gateway. Each gateway and the terminals connected to it (cell members) form a cell. This scenario finds important practical examples in cellular networks, Wireless LANs, some military networks (where low complexity nodes, like soldier radios, are directly connected to more sophisticated terminals, like Command & Control Centers, airplanes and so on), last mile connectivity or mesh networks (where mesh node communication may be supported by means of a different wireless technology than that of cell members).

Phoenix and two benchmark protocols are described in Section II, while Section III reports the actual performance evaluation and shows that Phoenix is able to improve throughput, cell capacity and energy efficiency by 10%-30% over conventional CSMA and decode-and-forward cooperation.

II. PROTOCOLS DESCRIPTION

In order to evaluate the performance of Phoenix, we consider two benchmark protocols. The former is the basic

1Note that MIMO_NC does not need multiple antennas [9], [10]; throughout this paper each node is assumed to have a single antenna.
IEEE 802.11 CSMA medium access policy without channel negotiation [11], which is based on carrier sensing and Binary Exponential Backoff. Such protocol will be referred to as CSMA in the remainder of the paper. The latter is an extension of CSMA (called Cooperative CSMA or CCSMA), whose main improvement over plain CSMA is decode-and-forward cooperative ARQ. While the mechanism of conventional CSMA is well known and does not need further explanation, CCSMA and Phoenix have been extensively discussed in [10]. This Section briefly outlines their main properties and highlight the differences with [10].

A. CCSMA: A Cooperative CSMA

In this scheme, cooperation is employed to improve ARQ reliability. Let us analyze CCSMA operation by means of the example in Fig. 1. A packet \( x \) sent by \( S \) fails to be delivered to \( D \); \( D \) replies by sending a Not ACKnowledgment (NACK) packet and by storing the corrupted version of \( x \) in its buffer so as to perform HARQ if a new copy of \( x \) arrives. The NACK packet triggers a distributed procedure to elect a relay that retransmits \( x \) on behalf of \( S \). The election phase works as follows:

- Each node that has correctly received both \( x \) and \( D \)'s NACK is a potential relay and starts a distributed contention phase to elect the actual relay. This criterion ensures that a contending node has both an error-free copy of \( x \) and is relatively close to \( D \) since it has correctly received the NACK.
- At the beginning of this phase, each potential relay picks a random backoff whose length \( n \) is uniformly drawn in the interval \([0, CW_{rel}]\).
- Only when the backoff phase is completed, does a potential relay check whether the aggregate received power is above or below the Carrier Sense threshold. In the former case, another neighboring transmission (possibly performed by the elected relay) is assumed to be going on, and the potential cooperator drops the contention. In the latter case, the node acts as relay and starts transmitting \( x \) to \( D \).

Upon receiving the new packet, \( D \) performs Chase combining with the previous copies of the same packet. If \( D \) correctly decodes \( x \), then an ACK is sent to \( S \) and the transmission is completed; otherwise, the source starts a new attempt until the maximum Short Retry Limit has been reached. Had no cooperative retransmission been performed, following the basic Binary Exponential Backoff strategy, the maximum backoff duration in slots would be twice as long as the maximum contention window of the previous phase. However, since the cooperative transmission takes always at least \( CW_{rel} \) slots, in CCSMA the new contention window is uniformly drawn in the interval \([0, CW_{backoff} - CW_{rel}]\). Cooperative procedures are triggered at the end of each attempt made by \( S \), as long as \( D \) does not decode the desired packet. The diversity order of the Packet Error Rate vs SNR is thus at least equal to the number of nodes involved in the transmission of \( x \).

CCSMA has the advantage of being a realistic cooperative system, because the distributed relay election phase can be easily implemented. Thus it is taken as a benchmark for decode-and-forward cooperative protocols.

B. Phoenix: Hybrid Cooperative-NC ARQ

Phoenix is a MAC protocol specifically tailored to maximize the potential of the MIMO_NC PHY. A distinguishing feature of MIMO_NC is the ability to retain corrupted or redundant coded frames to retrieve, together with other coded packets, the information encoded in them.

Phoenix is a random access protocol similar to CCSMA; however, the retransmission phase has to be slightly modified in order to exploit MIMO_NC. Let us refer again to Fig. 1. If a relay \( R \) has in its queue a packet \( y \) for \( D \), it may transmit a linear combination \( ax \oplus by \), where \( a \) and \( b \) are the coefficients used to combine the frames. In this case, MIMO_NC makes it possible for \( D \) to recover both \( x \) and \( y \) provided the SINRs of the two received packets are not too low. Because of the way such a PHY works, typically either both frames are recovered or none of them can be retrieved. If successful decoding takes place, \( D \) transmits two ACKs: one for \( S \) and another for \( R \). Otherwise, two NACKs will be transmitted. We notice that the decision statistics for \( x \) has a diversity order of 2, while for \( y \) it is 1. Hence, by taking advantage of MIMO_NC, Phoenix exploits the spatial diversity that motivates cooperation and at the same time increases throughput.

This paper also introduces an important improvement for Phoenix with respect to its original version. In [10], an elected relay would always perform a coded retransmission if it had a packet for \( D \). However, as discussed earlier, MIMO_NC cannot successfully decode the information bearing packets when the coded packet SINRs are too low. By evaluating the BER/SINR tables for MIMO_NC, it is possible to see that if the SINR of the first coded frame is below 3 dB (the NC threshold), the success probability is less than 66%, and thus MIMO_NC cannot operate with satisfactory reliability. Therefore, \( D \) signals to potential

---

\(^2\)The RTS/CTS negotiation present in [10] is not necessary in a clustered scenario, as all frames share the same destination address.
TABLE I
PARAMETERS USED IN OUR SIMULATIONS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission power</td>
<td>10 dBm</td>
</tr>
<tr>
<td>Noise Floor</td>
<td>-102 dBm</td>
</tr>
<tr>
<td>CS threshold, Detection threshold</td>
<td>-100, -96 dBm</td>
</tr>
<tr>
<td>Path loss exponent</td>
<td>3.5</td>
</tr>
<tr>
<td>Maximum Doppler shift</td>
<td>70 Hz (5 m/s)</td>
</tr>
<tr>
<td>Slot, DIFS, SIFS duration</td>
<td>20, 128, 28 μs</td>
</tr>
<tr>
<td>Data Rate</td>
<td>1 Mbit/s</td>
</tr>
<tr>
<td>Initial maximum contention window</td>
<td>128 slots</td>
</tr>
</tbody>
</table>

Simulation Time, Transient (metrics not collected) 30, 5 s
DATA header CSMA - CCSMA 272 bits
DATA header Phoenix 280 bits
Payload length 2000 bits
ACK/NACK 112 bits

Let us summarize the differences between CCSMA and Phoenix. First of all, they rely on different physical layers: Phoenix uses MIMO_NC while CCSMA conventional scalar detectors. We would like to stress that MIMO_NC places most of the computational burden at the receiver. In this setting, this means that the gateway has to be equipped with a more advanced PHY, but not its cell members. This is particularly important, because the gateway can afford more sophisticated radio circuitry and processing. Secondly, with our protocol, while performing a retransmission, a relay is also able to pursue its own interest, i.e., to deliver its own data, using the same transmission power and at no cost in bandwidth. This indeed motivates nodes to be relays, also because, as the next Section will prove, performing traditional cooperation does increase the overall network capacity but nodes which act as relays may degrade their own performance. The reason is the following. First, a relay must spend its own resources (time and energy) to carry somebody else’s data. Moreover, in a binary exponential backoff protocol (like the IEEE 802.11), a node resets the maximum contention window to the minimum value upon receiving an ACK. This implies that the terminal that has been helped will be more aggressive and thus may even transmit a new packet and hence capture the channel before the relay and in any case it will increase interference. These factors actually degrade the relays’ performance with respect to the case of no cooperation. Phoenix avoids the first problem and reduces the second [10].

III. SIMULATION RESULTS

The performance of the protocols described in Section II has been evaluated in clustered scenarios by means of extensive Omnet++ [12] simulations. We have considered networks composed by \( N_c \) non-overlapping cells of radius 75 m; each of them is made up by a gateway (GW) located at the center and \( n \) additional nodes that generate single hop traffic flows addressed to the GW. The transmitters are randomly distributed within an annulus centered at the GW of inner and outer radii equal to 25 m and 75 m, respectively. Such a configuration has been chosen to reduce the number of nodes very close to the GW. CSMA with Binary Exponential Backoff is known to lead to severe unfairness, where nodes in favorable positions get the vast majority of the bandwidth. This situation is not interesting, because that would basically turn off peripherals terminals and leave only the central nodes active. Furthermore, only the uplink is studied, because NC retransmissions offer no clear advantage for downlink communications. In order to understand why, consider the case of a cell member A not successfully receiving a packet from the GW. Another cell member B that acted as relay would have no packet in queue for A, as all its traffic is addressed to the GW. Hence, in a clustered scenario, users within a cell are not able to exploit the coded retransmission mechanism of Phoenix. Moreover, even if the GW were allowed to act as a relay for itself and to perform a NC retransmission, the two coded packets would be sent by the same node (the GW), thus offering no spatial diversity. For these reasons, we believe that the downlink would benefit in no special way from NC retransmissions.

In our simulations, all the nodes in the network share the same bandwidth (i.e., universal frequency reuse).

The standard set of parameters used for the protocols is reported in Tab. I. The protocols have first been evaluated in a reference scenario, with \( N_c = 4 \) and \( n = 9 \) (Section III-A). The Short Retry Limit has been set to 3, a suitable value for delay constrained applications. In these conditions the network is heavy loaded, and transmissions tend to be affected by a high level of both intra-cell and inter-cell interference (as no frequency division multiplexing has been considered). This setting is critical for CSMA-based medium access policies, and represents a good test for CCSMA and Phoenix, as the high number of packet losses is likely to often trigger cooperative procedures.

We have also studied how the performance of Phoenix and its competitors depends on certain design parameters and the scenario. For the former aspect we have investigated, in particular, how the number of retransmissions (SRL) affects the performance of the protocols. As far as network topologies are concerned, we have performed simulations varying both the number of users per cell and the number of cells in the network. The results of these studies are discussed in Section III-B.

In all cases, the wireless environment is subject to correlated Rayleigh fading. The Doppler frequency is equal to 70...
Hz, corresponding to a speed of 5 m/s at 2.4 GHz. All the results that we discuss in the remainder of the Section have been obtained by averaging the outcomes of 25 independent simulations.

A. Reference Scenario \((N_c = 4, \ n = 9)\)

The first metric that we consider is the average Packet Delivery Ratio (PDR), defined as the ratio of the number of packets successfully received at the GWs to the number of packets injected in the network (i.e., transmitted at least once, either by means of a direct transmission or by means of an NC phase). The PDR is depicted against the nominal average generated load \(\lambda\), in pk/s at every node, in Fig. 2.

First of all, we notice that both CCSMA and Phoenix are able to significantly improve reliability with respect to CSMA when operated at the same SRL. The reason for this is that when a GW is not able to successfully receive a packet, CSMA can only rely on the temporal diversity provided by successive retransmissions in order to recover the failure. On the contrary, protocols that implement cooperative relaying are able to take advantage of spatial diversity as well, which strongly enhances the probability of successfully performing ARQ phases and thus improves the overall PDR. All the curves in Fig. 2 mildly rise for high loads. This effect is a result of the well known unfairness that affects CSMA-based protocols. In every cell, nodes that are closer to the GW experience a success ratio higher than the one of terminals that are farther away. As the load increases, the former nodes tend to access the channel more often, while the latter get stuck for longer periods in backoff cycles due to their lower success probability and deliver fewer packets. For these reasons, terminals closer to the GW have a larger weight on the overall metric computation, thereby inducing the observed PDR rise. It is interesting to remark that this effect is far less pronounced in CCSMA and Phoenix. Cooperative relaying helps in particular nodes that incur frequent packet losses, enhancing their success probability and thus shortening the number of backoff phases they have to undergo. Therefore, this technique is extremely beneficial to reduce unfairness. Finally, we notice that Phoenix yields a slight improvement over CCSMA (around 3% at high loads). This effect stems from the higher reliability of cooperative-NC phases. As discussed in Section II, a GW that asks for a retransmission of a packet \(x\) allows cooperators to perform NC only if \(x\)’s SINR is higher than the NC threshold. This policy, thus, leads to a high probability of success for data sent by means of network coding and increases the overall average PDR.

Aggregate network throughput is depicted in Fig. 3 against \(\lambda\). The plot shows that Phoenix outperforms CSMA by 16% and CCSMA by 10%. The improvement offered by our protocol is twofold. On the one hand Phoenix takes advantage of cooperative relaying in order to both increase the number of successfully delivered packets and to reduce the time required to perform a communication. On the other hand, cooperative transmissions can be exploited by nodes that act as relays to opportunistically deliver their own traffic without negotiating the channel. The combination of these two factors significantly increases the number of served and delivered packets.

In order to get further insight on the performance of CCSMA and Phoenix, it is interesting to consider the metrics depicted in Fig. 4: impact of cooperative phases and impact of NC phases. The former is computed as the ratio of the number of packets sent by relay nodes (either as pure cooperation or as NC combinations) to the number of NACKs sent out by the GWs asking for a retransmission, while the latter is the ratio of the times MIMO_NC is actually used to the same denominator. At low loads, cooperative phases take place with high probability when a retransmission has to be performed, proving the effectiveness of the contention scheme implemented by CCSMA and Phoenix. On the
other hand, MIMO_NC is very rarely used, as nodes that act as relays are unlikely to have own packets to send to the GW. These two remarks explain both why at low loads the cooperative protocols are able to obtain interesting throughput gains with respect to CSMA and why Phoenix does not improve over CCSMA. As the load increases, the impact of cooperative phases tends to decrease, stabilizing to 50%. This is due to the higher level of interference, that may on the one hand prevent some nodes from decoding the original packet or the NACK sent by the GW (thus reducing the number of potential cooperators) and on the other hand may induce relay candidates to erroneously leave the contention phase (see Section II). The impact of NC phases, instead, increases with load as expected, because of the larger queue sizes. In saturation, a cooperative transmission involves a MIMO_NC encoded packet in the vast majority (75%) of cases. This behavior is reflected once again in Fig. 3: for high traffic rates Phoenix significantly outperforms CCSMA as the benefits of MIMO_NC become more and more important.

Another metric of interest is transmit energy consumption, presented in Fig. 5. The metric is computed as the total energy spent for packet transmissions divided by the number of successfully delivered information bits. The plot shows that Phoenix is more energy efficient than its competitors: at high loads, CSMA is outperformed by 20% while the improvement over CCSMA is as high as 10%. This stems from the capability of our protocol to exploit retransmission phases in order to deliver information packets at no additional cost in terms of energy and bandwidth. Two more observations can be made: first of all, the curves in Fig. 5 tend to decrease for higher loads, as an effect of the unfairness of CSMA-based protocols discussed earlier. Secondly, the gain of Phoenix over CCSMA is strengthened as traffic rate increases. This is due to the higher impact of NC phases on the cooperative mechanism that characterizes high load conditions (Fig. 4).

In our studies we have also investigated the behavior of the protocols with respect to network capacity. We define $\lambda$ as the minimum value of $\lambda$ that saturates the bandwidth $B$, i.e., $\lambda n T_p = B$, where $T_p$ is the minimum time to complete a data exchange (including overhead). Moreover, we call $L$ the payload size and define $\bar{\tau} = \lambda L$, the target throughput per node corresponding to this condition. In the reference scenario, $\lambda \simeq 40$ pk/s per node and $\bar{\tau} = 80$ kb/s per node. We identify four classes of terminals with respect to QoS; the highest class contains nodes that achieve an average throughput $\tau$ higher than or equal to $\bar{\tau}$; the second class is for cell members that satisfy the constraint $2\bar{\tau}/3 \leq \tau < \bar{\tau}$; the third class groups terminals that obtain a throughput $\bar{\tau}/3 \leq \tau < 2\bar{\tau}/3$ and finally the lowest class includes nodes that do not reach a minimum target throughput equal to $\bar{\tau}/3$. The results obtained at high load (60 pk/s) for the three highest classes are reported in Fig. 6. The improvement offered by Phoenix over its competitors is twofold. On the one hand, the number of cell members
that support the minimum target throughput increases by 9% with respect to CCSMA and by 20% with respect to CSMA. On the other hand, our protocol boosts the number of nodes with medium and high QoS by as much as 35% if compared to the other medium access policies. These results show that the combination of cooperative relaying and network coding is able to guarantee a minimum service to a much higher share of cell members thanks to a more fair distribution of the resources. Moreover, MIMO_NC may be extremely beneficial for applications characterized by high QoS constraints, as networks that rely on Phoenix can support many more cell members with such requirements even in harsh interference conditions.

Let us now focus on the trends for CCSMA and CSMA in Fig. 6. Two behaviors can be observed: i) CCSMA increases the number of nodes that support the minimum throughput service, i.e., \( \tau \geq \bar{\tau}/3 \); ii) the cardinality of the two highest QoS classes for CCSMA is slightly lower than the one that characterizes CSMA. This offers an interesting insight on the impact of cooperative relaying techniques. In a decode-and-forward approach, relay nodes spend some of their resources in order to help other terminals. In this way, not only do cooperators reduce their performance, but also nodes that benefit from relaying become more aggressive, as the enhancement of their success rate leads them to contend for the channel more often. Both these factors have a detrimental effect on cell members that are more likely to cooperate, i.e., those that are closer to their GW and that would normally enjoy high performance. 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, does not affect Phoenix: the MIMO_NC scheme that we propose does not penalize terminals that decide to act as relays but rather it boosts their performance by letting them exploit cooperative phases to serve their own traffic. Thus, our approach overcomes an important limit of cooperative relaying, improving the performance of both high and low QoS classes.

In order to complete our analysis of network capacity, we have investigated how the number of nodes with poorest performance depends on the specific protocol. To this aim, we consider an outage metric, defined as the 10-th percentile for the throughput of terminals in the network. The results are plotted in Fig. 7 for nominal loads up to 40 pk/s. As expected, cooperation strongly enhances the performance of low QoS cell members, with improvements between 30% and 50% with respect to CSMA at low and intermediate traffic rates. As the load increases, we observe a flattening of the outage throughput for all the protocols. If we combine this result with the outcome of Fig. 6, we can infer that while cooperative relaying and MIMO_NC do increase the percentage of users that enjoy a reasonable QoS, a certain percentage of nodes in the network always perform poorly due to their unfavorable position with respect to their GW. Fig. 7 also shows that Phoenix outperforms CCSMA in terms of outage throughput by as much as 15% at intermediate loads. This stems from the beneficial effect that MIMO_NC has on all the terminals in the network: whenever a node performs an NC transmission, it also releases the bandwidth that it would have needed to serve its own traffic it included in the combined packet. These resources can then be exploited by other cell members, improving the overall fairness (and thus the outage throughput as well).

In conclusion, we stress that the important improvements achieved by Phoenix in terms of capacity are quite general and scale well with the network size. This can be seen in Fig. 8, where the QoS distribution for the users is reported for a network of 10 cells, and 10 nodes per cell (i.e., 100 overall terminals). The percentages and results discussed for Fig. 6 are confirmed also in this larger scale scenario.

### B. Parametric Studies

In our work, we have also studied the performance of the considered medium access strategies with different
parameters and in various scenarios. First of all, we have investigated the impact of the SRL value on Phoenix and its competitors. The protocols were operated in the reference topology, with $N_c = 4$ and $n = 9$. The simulation results for aggregate throughput, energy consumption and network capacity are reported at high load (60 pk/s) in Tab. II. As a general remark, we notice that the gains of CCSMA and Phoenix over CSMA become less pronounced as the number of retransmissions increases. For instance, the throughput improvement of Phoenix with respect to CSMA lowers from 16% (SRL 3) to 11% (SRL 5), while energy saving drops from 19% (SRL 3) to 7% (SRL 5). This is an effect of the longer backoff phases that nodes experience for higher SRL values. A longer average interval between two successive retransmissions, in fact, increases the temporal diversity at the receiver. Therefore, a protocol that only relies on this form of diversity to recover packet losses (e.g., CSMA) significantly improves by raising the SRL parameter. As a consequence, the impact of spatial diversity is reduced, affecting the effectiveness of both CCSMA and Phoenix. From the parametric analysis reported in Tab. II we can then draw two conclusions. First, our protocol offers interesting improvements regardless of the SRL value. Second, Phoenix turns out to be particularly effective for delay constrained applications, when few retransmissions can be tolerated.

Then, we have tested the considered protocols for different cell densities. This study is of interest to analyze the robustness of our MAC schemes, as an increase in the number of transmitters within a cluster leads to harsher conditions for both interference and channel contention. Fig. 9 reports the aggregate throughput for low (20 pk/s) and high (60 pk/s) loads, depicted against the number of nodes per cell. At low traffic rates, the metric increases with the number of transmitters, as expected. On the contrary, when the network reaches saturation, the throughput does not scale significantly with the number of nodes. This is due to the lower success rate induced by interference that characterizes denser scenarios. This effect is less pronounced for Phoenix, whose gains with respect to the competitors rise with the number of transmitters. At the maximum density, our protocol outperforms CSMA by 17% and CCSMA by as much as 13%. The capability of opportunistically exploiting cooperative phases to serve additional traffic turns out to be extremely beneficial when the network is congested. Phoenix, in conclusion, offers a much better scaling with the number of transmitters, as expected. On the contrary, the number of transmitters within a cell, the value of the target throughput $\bar{\tau}$ varies, as shown by the definition in Section III.A.

\[ \text{Aggregate throughput [kb/s]} \]

<table>
<thead>
<tr>
<th>SRL</th>
<th>Throughput [kb/s]</th>
<th>CSMA</th>
<th>CCSMA</th>
<th>Phoenix</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1128</td>
<td>1202</td>
<td>1320</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1259</td>
<td>1308</td>
<td>1420</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1354</td>
<td>1386</td>
<td>1502</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Energy [nJ/bit]</th>
<th>Capacities:</th>
<th>Percent users in the two highest QoS classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>19.33</td>
<td>18.89</td>
</tr>
<tr>
<td>4</td>
<td>23.90</td>
<td>25.4</td>
</tr>
<tr>
<td>5</td>
<td>28.20</td>
<td>29.50</td>
</tr>
</tbody>
</table>

\[ \text{Capacity:} \]

\[ \text{Performance dependence on node density} \]

4 Cells, SRL = 3, Values of $\lambda$ chosen to induce saturation

<table>
<thead>
<tr>
<th>nodes/cell ( $\lambda$ [pk/s] )</th>
<th>CSMA</th>
<th>CCSMA</th>
<th>Phoenix</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 (80)</td>
<td>23.94</td>
<td>21.61</td>
<td>20.40</td>
</tr>
<tr>
<td>8 (70)</td>
<td>25.03</td>
<td>23.08</td>
<td>21.65</td>
</tr>
<tr>
<td>10 (60)</td>
<td>27.57</td>
<td>25.30</td>
<td>23.18</td>
</tr>
<tr>
<td>12 (60)</td>
<td>28.03</td>
<td>26.06</td>
<td>23.72</td>
</tr>
</tbody>
</table>

\[ \text{Energy [nJ/bit]} \]

\[ \text{Capacity:} \]

\[ \text{Percentage users in the two highest QoS classes} \]

\[ \text{Performance dependence on short retry limit} \]

4 Cells, 10 nodes/cell, $\lambda = 60$ pk/s

\[ \text{Table II} \]

\[ \text{Table III} \]
As the number of clusters increases, more simultaneous transmissions can take place, but the inter-cell interference grows. The former factor suggests beneficial effects on aggregate metrics due to better parallelism. On the contrary, the latter should be detrimental for per-node metrics, as the average SINR reduction would adversely affect the reliability of single links. These predictions are confirmed by the results of our simulations. In particular, the first effect is represented in Tab. IV: the number of successful communications increases with the number of cells because of additional parallelism. Therefore, the aggregate throughput grows and the energy consumption per information bit decreases. Tab. IV also shows that Phoenix again outperforms the other protocols in terms of the considered metrics.

Instead, the impact of the second phenomenon can be seen by looking at network capacity (that stems from per-node throughput). This is reported in Fig. 10 as the percentage of cell members in the two highest QoS classes for the different protocols against the number of clusters. As discussed, the poorer link quality due to inter-cell interference negatively affects the performance of cell members. This is especially detrimental for users in high QoS classes and induces a degradation of the depicted metric.

Two further remarks can be made on Fig. 10. First of all, the trends discussed for the reference scenario (see Fig. 6) are confirmed: cooperative relaying redistributes the resources in the network at the expense of users that would enjoy high performance. On the contrary, with MIMO_NC, the average performance of nodes can be boosted without affecting terminals with high QoS. Secondly, Fig. 10 shows that the percentage gains of Phoenix with respect to CCSMA slightly reduce as the number of cells increases. This effect can again be explained considering how the higher inter-cell interference lowers the average SINR at a GW. Recalling that NC may be triggered only if minimum quality conditions are met at a GW for a non-correctly received packet, it follows that the percentage of MIMO_NC transmissions over the cooperative phases decreases with the rise of background interference. We can then infer that for a high number of lightly loaded cells, Phoenix tends to approach CCSMA’s behavior, as confirmed also by the results of Tab. IV.

IV. CONCLUSIONS

This paper has carried out an extensive performance evaluation of Phoenix in clustered networks. It has been shown that interesting performance improvements in terms of throughput and especially capacity can be achieved for a wide range of design parameters. The dependence of the performance on these parameters has been analyzed and we have found that Phoenix is particularly suited for high density networks with tight delay requirements.

REFERENCES