Network-Aware Retransmission Strategy Selection in Ad Hoc Wireless Networks

Federico Librino*, Giorgio Quer®, Michele Zorzi§

*Italian National Research Council, Via Moruzzi 1 – 56124 Pisa, Italy
®University of California San Diego – La Jolla, CA 92093, USA
§DEI, University of Padova, via Gradenigo 6/B – 35131, Padova, Italy

Abstract—Retransmission strategies have been widely investigated in wireless networks, since they are able to grant considerable benefits in dynamic environments. Distributed schemes, based on cooperative techniques, can also add the benefits of spatial diversity, particularly if combined with Multi-User Detection decoding schemes. However, the impact of each scheme on the rest of the network cannot be neglected, since it also affects the overall network performance. A mathematical approach to evaluate this impact can be very involved, given the potentially very large number of parameters to take into account. In this paper, we propose instead a probabilistic approach, based on Bayesian Networks, to determine the expected impact, in terms of interference, of different schemes on the rest of the network. Through our framework, it is possible to adaptively select the best scheme to use, as a function of the observation of some topological parameters. We also design a distributed protocol to implement a variety of retransmission schemes, and the performance results confirm the effectiveness of our model over a static choice of the retransmission strategy and also over a selfish retransmission scheme that always selects the strategy that maximizes the probability of success of the retransmission.

I. INTRODUCTION AND RELATED WORK

The design of smart and efficient communication schemes over wireless networks is challenging. The inherent broadcast nature of the medium, as well as the dynamic environment, can severely impact the transmission reliability. In infrastructured networks, where the interference is limited, the variations in channel conditions can be countered by specific improvements in both physical and MAC layers. Ad hoc networks are instead more susceptible to link failures, due to the lower level of coordination among nodes, which commonly leads to an increased interference level.

Adaptive strategies [1]–[3] have been long studied as a viable solution to improve the link quality when and where necessary. Proactive strategies usually make use of channel information to appropriately tune some key parameters, such as the transmission power or the communication rate. On the other side, reactive strategies usually rely on feedback from the destination node, which can trigger a second transmission of the lost data. ARQ (Automatic Repeat reQuest) schemes were among the earliest to be developed, and have been widely investigated, in terms of throughput performance and energy efficiency, in a wide set of scenarios, see [4].

These schemes can be improved if nodes are collaborative. In this case, nodes that are close to the source and the destination can be elected to perform retransmissions on behalf of the source, thus exploiting better channel conditions. Relaying is the basis of multi-hop networks and can be seen, in fact, as a form of cooperation. Several protocols have been designed, in order to balance the advantages offered by the relays and the increased interference level due to the higher number of total transmissions [5]–[7]. Cooperation has also gained attention in recent years, being recognized as a promising way to increase the network performance in terms of throughput, latency and energy consumption. In particular, Coded Cooperation (CC), due to its flexibility, was shown to be suitable for ad hoc networks. In CC, the cooperators provide the destination with additional redundancy, rather than with identical copies of the lost packet, which results in an adaptively varying coding rate. Protocols based on this scheme can be found in [8], [9] and, combined with HARQ, in [10]. In this work, terminals can exploit Multiuser Detection (MUD) and Successive Interference Cancellation (SIC), thus avoiding the need to choose a single cooperator per transmission.

The problem of most of these approaches is that a network perspective is missing. Indeed, a retransmission may be beneficial to a subset of nodes, but at the same time it may raise the interference level and severely impact other concurrent communications among surrounding nodes, resulting in a reduced overall performance.

To determine the impact of a given strategy on the rest of the network is not an easy task, since it depends on the transmission parameters, the implemented protocol and the network topology. In this paper, we propose a heuristic metric to measure how the usage of a given retransmission strategy can affect the rest of the network. The probabilistic relationships between this metric and some local topological parameters is obtained through a Bayesian Network (BN) approach. A BN is a probabilistic graphical model [11] that is used to learn the probabilistic relationships among a set of variables, as a function of a finite dataset of outcomes of the same variables. A BN approach similar to the one proposed in this paper has been used in [12] to optimize the inter-network cooperation between ad hoc wireless networks. BNs are also exploited in [13] to optimize some broadcasting mechanisms in a Mobile Ad Hoc Network, where each node can adaptively tune some key parameters of a predefined protocol or switch between different broadcast protocols. In this paper, the BN approach is exploited to design a protocol which can employ different retransmission schemes, with the aim of successfully delivering the data packet with limited impact on the rest of the network.

The rest of the paper is organized as follows. In Sec. II we detail the system model and the protocol implementation, then in Sec. III we describe the BN learning phase and the strategy selection scheme. The performance results are highlighted in Sec. IV, and Sec.V concludes the paper.

II. SYSTEM MODEL

The proposed approach can be applied to a wide variety of scenarios, since the only requirement is to identify a set of
strategies to be analyzed, a global performance metric, and a set of parameters which affect the overall network behavior. In the following, we restrict our attention to a specific multi-hop network scenario, in which we have studied via a realistic simulation the effectiveness of several retransmission strategies and of our own retransmission strategy selection scheme.

A. Scenario

We focus on a homogeneous ad hoc network with \( N \) nodes randomly deployed in a square area of fixed size. Each node is a source of information packets, whose destinations are randomly chosen among the other \( N - 1 \) nodes. Packets at each node are generated according to a Poisson random process of intensity \( \lambda \). Time is slotted, and we assume that slot synchronization is granted, although schemes for non-synchronized networks can be investigated as well.

The transmission power \( P \) is the same for all nodes. The channel between each pair of nodes is affected by Rayleigh fading, which is assumed to be: 1) constant within a single time slot, and 2) time-correlated between consecutive slots, with correlation factor \( \rho = 0.9 \). Therefore, the Signal to Noise Ratio (SNR) in time slot \( t \) between two nodes \( i \) and \( j \), located at a distance \( d_{ij} \) from each other, is given by:

\[
\Gamma_{ij} = \frac{P}{AN_0} d_{ij}^{-\alpha} |h_{ij}(t)|^2,
\]

where \( N_0 \) is the noise power, \( \alpha \) is the path loss exponent, \( A \) is a constant and \( h_{ij}(t) \) is a complex Gaussian random variable with zero mean and unit variance.

Since all nodes use the same spectrum, multiplexing is achieved via CDMA: a unique spreading sequence is assigned to each node, with a spreading factor \( N_s \). These sequences are not necessarily orthogonal, meaning that concurrent transmissions may interfere with each other. As a consequence, with this CDMA scheme, we can express the Signal to Interference plus Noise Ratio (SINR) between nodes \( i \) and \( j \) as:

\[
\Lambda_{ij} = \frac{N_s \Gamma_{ij}}{\sum_{k \neq ij} \Gamma_{kj} + 1},
\]

where \( k \) spans over all the other nodes which are transmitting in the same time slot. The modulation used is BPSK, resulting in a packet error probability:

\[
P_e = 1 - (1 - Q(\Lambda_{ij}))^\ell,
\]

where \( \ell \) is the packet length in number of bits, and \( Q(\cdot) \) is the Gaussian complementary cumulative distribution function.

Since there is no centralized control, interference can severely impact the decoding probability. A successive interference cancellation (SIC) scheme may be applied to achieve Multiuser Detection (MUD). In this case, when two or more signals are received in the same time slot, they are first sorted in decreasing order as a function of their SINR. Subsequently, the receiver attempts to decode the signal with the highest SINR. If it is successful, the signal is canceled, and the remaining SINRs are recomputed. Then, the process is iterated until all the signals of interest have been considered.

At the MAC layer, before each transmission, there is an RTS/CTS exchange, in order to probe the destination availability. Both the RTS and the CTS are very short, and are coded and transmitted over a single time slot. A single data packet is fragmented into \( N_b \) blocks, and each block is sent over a single time slot, with a short CRC attached. Thus the transmission of a packet spans \( N_b \) time slots. The value of \( N_b \) should be accurately tuned depending on the considered scenario: large values are useful to reduce the required handshaking overhead, but may in turn result in a lower packet success probability. After the packet transmission, the intended destination sends an ACK, where it declares whether the packet has been received or not, specifying which blocks of the packet are still missing, if any. If the packet was not correctly decoded, a retransmission occurs.

Furthermore, when the distance between the source and the destination exceeds the maximum link length \( d_0 \), we have a multi-hop transmission. Assuming that each node is aware of its geographic location, it determines its routing table by using a standard OLSR algorithm. When a node receives a packet to be forwarded, the packet is inserted in the FIFO queue of packets to be transmitted, with a maximum buffer length equal to \( M_b \).

B. Set of retransmission strategies

After the transmission, the destination node sends an ACK packet to the source specifying the packet fragments still missing. Several retransmission schemes can be applied, which leverage on time or space diversity, as well as on the MUD capability of the receiver. The key factors to define such schemes are 1) which nodes should perform the retransmission; and 2) which information should be included in the retransmission.

As to the first factor, we investigate three options: the retransmission can be handled by the source node itself (strategy \( S \)), by a subset of nodes, called cooperators (strategy \( CC \)) or by a selected node among the available cooperators (strategy \( SC \)). The first choice exploits the time diversity offered by the channel, while the second and the third rely on spatial diversity. On the one side, a cooperator is likely to offer an increased probability of success, but on the other side, the presence of nodes available to cooperate is not guaranteed.

When the retransmitting node is selected, the amount of data to be retransmitted has to be determined. In this case, we investigated two options. Since information about the missing fragments is broadcasted in the ACK packet, retransmitting the whole packet is not necessary. Nevertheless, \( N_{ts} \) time slots may still be exploited to retransmit the missing fragments more than once. If, for instance, \( N_{b0} \leq N_{ts} \) fragments were lost, they could be retransmitted in \( N_{ts} \) time slots (with repetitions), thus increasing the probability of recovering them. This choice (strategy all, in which all the \( N_{ts} \) time slots are exploited) is useful if the scenario requires all the transmitted packets to be of the same length. In addition, the repeated transmission of the missing fragments can be seen as a form of coding, which aims at improving the communication robustness\(^1\). The risk of wasting resources is balanced by the fact that a failed retransmission would result in an entire new communication to be established after a random backoff time, thus significantly increasing the delivery delay. A more aggressive option uses only \( N_{b0} \) slots to transmit the \( N_{b0} \) missing fragments (strategy \( red \), in which a reduced number of time slots is exploited). This strategy requires a lower amount of resources, but also reduces the decoding probability of the remaining fragments, and therefore may not always be the best choice. The set of strategies are reported in Tab. I.

\(^1\) More advanced coding schemes may be used as well, to further boost the protocol performance. We leave this as a topic for future work.
TABLE I
THE SIX RETRANSMISSION STRATEGIES AS A FUNCTION OF THE CHOSEN RETRANSMITTING NODES AND THE NUMBER OF TIME SLOTS USED.

<table>
<thead>
<tr>
<th>Source tx</th>
<th>Set of coops tx</th>
<th>Selected coop tx</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-all</td>
<td>S-all</td>
<td>SC-red</td>
</tr>
<tr>
<td>S-red</td>
<td>CC-red</td>
<td>SC-all</td>
</tr>
</tbody>
</table>

C. Protocol implementation

Each retransmission strategy aims at recovering packets not successfully decoded. The time-varying network environment, where channel conditions and interference levels may be highly dynamic, suggests that an adaptive approach would bring the most benefit. In particular, the protocol implementation should be very flexible: it should first collect the parameters of interest and then select the best retransmission strategy as a function of such parameters. The selected strategy must then be broadcast by the source node.

In line with common cooperative retransmission implementations, whenever an ACK declaring a decoding failure is received, both the source node $S$ and the destination node $D$ keep listening to the channel for an additional time slot. This time slot is reserved for the idle surrounding nodes which, upon overhearing the negative ACK, declare their availability to cooperate. All the available cooperators therefore concurrently transmit an AVA (available to cooperate) packet to $S$, where they also add information on their channel conditions towards $D$ (estimated when decoding the ACK). Node $S$ is therefore informed about the number of available cooperators and the estimated quality of their channels. The best cooperator is considered to be the one which has the best channel conditions towards the destination. Note that this information may not be fully reliable, if based on the SINR, since the interference level at the cooperator is likely different from that at $D$. For a more accurate selection, we could instead rely on information explicitly sent by $D$. In fact, the interference level at a given time $t$ is likely to be correlated to that at time $t - 1$, and is also dependent on topological or network parameters (number of neighbors, traffic type, etc.). Therefore, if the interference level is also included in the ACK, and a dynamic probabilistic model can predict its variations, the choice of the best cooperator may be more effective. This analysis is beyond the scope of the paper, and we leave it as a subject for future investigation.

Having collected relevant information about the channel conditions and the surrounding nodes, node $S$ can choose the best retransmission strategy. Then, in the following time slot, the source node broadcasts a scheduling packet SCH, to inform both the destination and the set of available cooperators about the selected strategy. This short packet, as the other control packets, is protected through a rate $1/2$ convolutional code, to grant a high decoding probability at $D$. The retransmission then takes place, followed by a new ACK from the destination. If some fragments are still missing, the source schedules a new transmission attempt, with a new handshake procedure, after a random backoff time (for simplicity, only one retransmission is allowed). After $M_{tx}$ failed attempts, the packet is finally discarded. As an example, a graphical representation is depicted for strategy SC-red in Fig. 1; analogous schemes can be drawn also for the other strategies.

III. RETRANSMISSION STRATEGY SELECTION

In this section we detail the network information available at the transmitting node, and we represent such information as a set of variables that can change in each time slot. Then, we summarize the concept of Bayesian Network (BN), a probabilistic graphical model that is exploited to learn the probabilistic relationships among these variables. The BN is then used to infer which of the six retransmission strategies to select in order to optimize the performance of the network.

A. Network information available

When one or more of the transmitted fragments are not correctly received, the transmitting node $S$ should select a retransmission strategy based on the information collected before the retransmission. This information is collected through the reception of the ACK and AVA packets, and includes also the topological information available to the node. The useful information contained in the ACK can be represented by the following variables:

- $\pi_1 = N_0$, the number of fragments not correctly received at the destination $D$;
- $\pi_2$, the channel coefficient between $S$ and $D$ during the transmission of the last fragment;
- $\pi_3$, the SINR at $D$ during the transmission of the last fragment.

The useful information collected thanks to the reception of the AVA packets can be represented by:

- $\pi_4$, the number of available cooperators;
- $\pi_5$, the number of neighbors of the cooperators;
- $\pi_6$, the channel of the cooperator with the best channel towards $D$ (this parameter can be estimated by the cooperators during the transmission of the ACK).

Furthermore, we assume that node $S$ knows the number of its own neighbors ($\pi_7$). The set of all the parameters collected before the retransmission is named $\Pi = \{\pi_1, \ldots, \pi_7\}$.

The goal of the BN analysis is to find the probabilistic relationships among the parameters in $\Pi$, and the retransmission performance, as a function of the adopted strategy. The set of performance parameters can also be collected after the retransmission, and includes $N_{f1}$ and $q$. $N_{f1}$ is the number of fragments not correctly decoded after the retransmission; we highlight that inferring the value of $N_{f1}$ before performing the retransmission is very involved when interference is taken into account. $q$ is the network impact, defined as the amount of interference that the selected strategy causes to the rest of the network, and can be calculated as the sum of the average power that is received by the nodes in the set $N$, defined as the union of the sets of the neighbors of all the nodes in $C$, i.e., the nodes which are transmitting during the retransmission phase\(^2\). The set $N$ does not include $S$, $D$ and

\(^2\)While we focus on these two metrics in this paper, our framework can easily accommodate other metrics as well (e.g., energy of packet delivery ratio).
all the nodes which are participating as cooperators. The sum is finally multiplied by the duration (in number of slots) of the retransmission phase, since a longer retransmission will cause a higher damage to other ongoing communications. Therefore, if \( D_r \) is the duration (in number of slots) of the retransmission phase, we can write:

\[
q = D_r \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{N}} P_{ij} \alpha_{ij},
\]

(4)

in the case of a retransmission involving cooperators, and

\[
q = D_r \sum_{j \in \mathcal{N}} P_{ij} \alpha_{ij},
\]

(5)

if the source node is in charge of retransmitting the missing fragments. Note that the value of \( q \) should be adjusted, since not all the surrounding nodes are receiving, during the retransmission phase, and idle or transmitting surrounding nodes are not affected by interference. To take this fact into account, we measure \( q \) by considering the interference perceived by the surrounding nodes only in the time slots they spend in data reception. In other words, the summations in (4) and (5) are computed, for each time slot \( k \), only over the subset \( \mathcal{N}_k \subset \mathcal{N} \) of data receiving nodes, and thus becomes:

\[
q = \frac{1}{D_r} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{N}_k} P_{ij} \alpha_{ij},
\]

(6)

The fading coefficients are not taken into account, since the obtained values of \( q \) are averaged at the end of the campaign set. We define the set of performance parameters that can be collected after the retransmission as

\[
\Phi = \{ N_{II}, q \}.
\]

We want to stress that, although the channel condition between the best cooperator and \( D \) (represented by \( n_b \)) is among the parameters in set \( \Pi \), we are not directly estimating the probability of successful retransmission based on the channel state measured after each failed transmission. Instead, we are using probabilistic relationships between the parameters in \( \Pi \) and the ones in \( \Phi \) which have been derived over a large amount of data.

B. Bayesian Network

A Bayesian Network (BN) is a probabilistic graphical model \([11]\) that can describe in a compact way the conditional dependence relations among a set of random variables, that are represented with a Directed Acyclic Graph (DAG). A DAG is a graphical representation of the conditional dependencies among the variables, that define the structure of the joint probability among these variables. In general, the graph representing the structure of the joint probability among a set of variables is a fully connected DAG.

The problem is that it is not possible to learn the probabilistic relationships among the variables based on a finite set of data with such a complex structure. Usually, some arbitrary assumptions allow the study of the joint probability. In our case, we exploit a structure learning technique to identify the presence of conditional independence conditions among the variables. If such conditions exist, then it is possible to simplify the DAG deleting some arrows and representing the joint probability with a simplified structure. The approach to select the BN structure includes a score based method \([11]\) and a Hill Climbing random search \([14]\), and can be summarized in the following steps:

- we select a valid (and random) BN structure (\( \text{DAG}_i \));
- we score this structure based on how well it represents the conditional distribution of the data in the training set, using the Bayesian Information Criterion (BIC) \([15]\);
- we select all the DAGs obtained from \( \text{DAG}_i \) by adding, deleting or changing the direction of a single arrow;
- we score each selected DAG and pick the one with the highest score, \( \text{DAG}^* \) (best fitting with the data);
- if \( \text{score}(\text{DAG}^*) > \text{score}(\text{DAG}_i) \), then \( \text{DAG}_i \leftarrow \text{DAG}^* \), and the process is repeated until convergence.

This technique can only find a local maximum of the best fitting DAG, so to improve the accuracy it should be repeated multiple times with different initial DAGs. The optimality can not be guaranteed, but this is a good technique to find local optima, and in our case we observed that they are close enough to the global optimum.

With the best fitting structure obtained as described above, it is possible to quantitatively learn the probabilistic relationships among the data, by means of a maximum likelihood parameter learning method. We refer to \([12]\) for further details on the specific choices made to learn a BN from the data and on the learning techniques involved.

C. Simulation setup: learning phase

In order to derive the probabilistic relationship between the parameters in \( \Pi \) and those in \( \Phi \), we set up a first simulation campaign to collect the data in the training set. For each scenario (in our case, for each value of the traffic load \( \lambda \)), we run a system level Matlab simulator over randomly generated network topologies, simulating the scenario described in Sec. II. We simulate 10000 time slots per topology: all the control packets are included in the simulation, which therefore takes into account also the overhead required by each strategy. Every time a retransmission is required, one of the six strategies is randomly chosen, with equal probability. In addition, an entry with the values of all the parameters in \( \Pi \) and \( \Phi \) is also recorded and stored in a database. The simulation campaign stops when 40000 entries are created and stored in the database.

D. Probabilistic relationships learned from the data

Based on the data collected during the learning phase, a BN describing the probabilistic relationships among the parameters in \( \Pi \) and \( \Phi \) can be learned. It will be used to infer the values of the \( \Phi \) parameters based on the current value of the \( \Pi \) parameters, thus suggesting the best retransmission strategy in terms of the expected values of the \( \Pi \) parameters.

We observed that the probabilistic relationships among the parameters change significantly as a function of the strategy selected, and of the amount of traffic in the network, \( \lambda \). Anyway, we can identify a subset of the parameters collected before the retransmission (\( \Pi_1 \subseteq \Pi \)) that separates each of the performance parameters (in \( \Phi \)) from the rest of the graph \( ^3 \), and this subset changes as a function of the strategy selected but is independent of \( \lambda \). Thus, we can study the probabilistic

\(^3\text{There is a subset of parameters in } \Pi \text{ that separates each of the parameter in } \Phi \text{ from the rest of the graph, according to the D-separation rules described in [16].}\)
relationships between the elements in this subset of parameters and the performance parameter. The subset $\Pi_1 \subseteq \Pi$ is represented in Tab. II as a function of the strategy selected and of the performance parameter of interest, i.e., $N_{t1}$ or $q$.

Observe Tab. II, we notice that $q$, when strategy $S$-all or $S$-red is selected, is statistically independent of the other parameters collected before the retransmission. We also notice, as expected, that the number of fragments still missing after the retransmission, $N_{t1}$, always depends on the number of missing fragments before the retransmission ($\pi_1 = N_{t0}$), for every choice of the retransmission strategy.

### E. Strategy selection

Each retransmission strategy, depending on the specific conditions, can grant a certain benefit in terms of delivery probability. At the same time, each strategy can be more or less harmful for the surrounding nodes of the network. We want the strategy selection to take into account both aspects, which are defined by means of the parameters $N_{t1}$ and $q$, described in Sec. III-A. We stress the fact that $N_{t1}$ and $q$ clearly depend on the strategy employed.

The choice between the various strategies has the primary objective of guaranteeing a minimum delivery probability $\gamma$. Besides this primary objective, we also aim at minimizing the interference impact on the surrounding communications. With the BN approach we can estimate the distribution of $N_{t1}(s)$ and the expected value of $q(s)$ as a function of the strategy $s$, with $s \in S = \{S$-all, $S$-red, $CC$-all, $CC$-red, $SC$-all, $SC$-red$\}$, and conditioned on the $\Pi$ parameters in the previous transmission.

We can hence define a new adaptive scheme, named $AD$, which works as follows: each time a retransmission is needed, the strategy is selected based on the parameters $\Pi$, collected at the source by means of the AVA packets. According to the expected values of $N_{t1}$ and $q$ predicted by the BN estimation, the chosen strategy is the one which solves the following minimization problem:

$$s^* = \arg \min_{s \in S} q(s)$$

subject to $p[N_{t1}(s) = 0] \geq \gamma$. \hspace{1cm} (8)

### IV. Results

#### A. Simulation setup

After obtaining the probabilistic relationship between the parameters of interest, as explained in Sec. III, we performed a second set of simulations. For each value of $\lambda$, corresponding to a given set of conditional probability distributions, we generated 100 random topologies, and for each topology we ran several simulations with a duration of 10000 time slots.

For each $s \in S$ we ran one simulation in which the same retransmission strategy was used throughout. In addition, we ran another simulation in which the adaptive scheme $AD$ was used instead.

### B. Performance comparison

We report here the performance results in terms of overall aggregated throughput. In our simulations, the network consists of $N = 30$ nodes, randomly deployed in a $200 \times 200$ $m^2$ square area. We plot the throughput as a function of the network load, in terms of $\lambda$, the packet generation intensity at each node, from $\lambda = 1$ ($\text{pkt/s}$), corresponding to a lightly loaded scenario, to 5, which represents a saturated network. The load is the key factor since it determines the interference level, whose effects can be mitigated by a proper adaptive scheme.

In Fig. 2, we show the aggregate throughput for the six retransmission schemes, as well as for the adaptive strategy based on the proposed BN approach (strategy $AD$). All the strategies are implemented through the protocol described in Sec. II, with each packet composed by $N_d = 6$ data fragments. It can be observed that, among the six base schemes, which one is the best depends on the traffic load. For low loads, cooperators are useful: not only are more nodes available to help, on average, but also a simultaneous transmission from multiple cooperators (strategies $CC$-all and $CC$-red) can be effective. Notice also that the schemes based on the retransmission of only the missing fragments perform worse, since the interference level makes it necessary to increase the time diversity to grant a good decoding probability. The adaptive strategy, however, performs always better, achieving almost a 10% gain over the best static scheme in all cases.

In Fig. 3 we consider the same scenario where a Successive Interference Cancellation (SIC) scheme is used to decode multiple incoming signals. The presence of MUD highly improves the communication robustness, since the effect of interference is reduced. Therefore, longer packets can be supported, so we increased the number $N_d$ of fragments per packet from 6 to 8. We observe that now the strategies based on the retransmission of the entire packet perform worse, since they not only increase the delivery delay, but also unnecessarily raises the overall interference level. Furthermore, the retransmission performed by the source is often preferable, since the usage of SIC makes...
the SINR change faster, and therefore enhances the benefit of time diversity. Furthermore, in this scenario, the adaptive strategy still achieves the best performance, although the gain is much lower, due to the reduced effect of interference among concurrent communications.

It may be argued that adaptive strategies can be designed with the aim of maximizing only the decoding probability of the current transmission. In scenarios where different communications are well separated (either by means of a controlled scheduling or because the topology allows it), such strategies can be optimized to get the best overall performance. However, when the retransmission has an impact on the rest of the network, more conservative schemes, such as the one proposed in this paper, are able to grant a higher performance in the worst case. In order to show this, we introduce another strategy, called $AD_s$, which is also adaptive but takes a more selfish approach: whenever a retransmission is needed, it behaves as $SC$-all if at least one cooperator is available, and as $S$-all otherwise. This strategy aims at exploiting both the spatial and temporal diversity, with no constraints on the generated amount of interference. We compare the results in Fig. 4. In a scenario with high load ($\lambda = 5$), we computed the aggregate throughput of the considered schemes over 100 topologies, and collected, for each strategy, the performance obtained in the worst case, normalized to the one of the selfish strategy $AD_s$. Nonetheless, we notice again that, although the $AD_s$ strategy performs better than the static ones, $AD$ can outperform it by about 10%.

V. CONCLUSIONS

In this paper we have proposed a probabilistic approach, based on Bayesian Networks, to infer the impact of different retransmission strategies, as a function of the network local conditions. We have also proposed an adaptive retransmission technique that exploits the information provided by the Bayesian Network to choose for each retransmission the best out of six strategies. Our results show that this technique outperforms all the static strategies, as well as an adaptive strategy that does not consider the impact of a retransmission on the rest of the network.

ACKNOWLEDGMENTS

This work was partially supported by the U.S. Army Research Office, Grant. No. W911NF-11-1-0338 and W911NF-11-1-0336, and by the CaRiPaRo Foundation, Padova, Italy, under the “Progetti di Eccellenza 2012” program.

REFERENCES