Inter-Network Cooperation exploiting Game Theory and Bayesian Networks

Giorgio Quer, *Member, IEEE*, Federico Librino, *Member, IEEE*, Luca Canzian, *Member, IEEE*, Leonardo Badia, *Member, IEEE*, and Michele Zorzi, *Fellow, IEEE*

***Abstract*—Relay sharing has been recently investigated to increase the performance of coexisting wireless multi–hop networks. In this paper, we analyze a scenario where two wireless ad hoc networks are willing to share some of their nodes, acting as relays, in order to gain benefits in terms of lower packet delivery delay and reduced loss probability. Bayesian network analysis is exploited to compute the probabilistic relationships between local parameters and overall performance, whereas the selection of the nodes to share is made by means of a game theoretic approach. Our results are then validated through the use of a system level simulator, which shows that an accurate selection of the shared nodes can significantly increase the performance gain with respect to a random selection scheme.**

***Index Terms*—Bayesian network, game theory, inter-network cooperation, multi-hop networks.**

I. Introduction

Cooperation is one of the most promising enabling techniques to meet the increasing rate demands and quality of service requirements in wireless networks, especially since nowadays many techniques to share the spectrum resources among different networks are envisioned [2]. Beyond spectrum sharing, also relay sharing is possible: namely, when a multi–hop network decides to cooperate, it shares some or all of its nodes, that become available as relays for another multi–hop network as well. In such a scenario, cooperation can leverage the benefits of multi–path diversity, since more paths connecting two nodes will be available, obtaining a considerable gain in the efficiency of shared resources. Sharing the whole set of nodes provides the highest number of paths available for each of the participating networks. However, this comes at the cost of increased traffic that should be handled by some of the shared nodes. In a realistic scenario, an operator may be willing to share only a (possibly small) subset of nodes, e.g., for security or privacy reasons. If this is the case, it becomes important to assess how the effectiveness of the cooperative scheme depends on how many nodes are shared and which ones, and to provide suitable selection schemes. Indeed, some nodes deployed in crucial positions may be particularly suited for helping the other network; on the contrary, nodes placed close to the network border are likely to be less useful or even useless. Furthermore, an operator sharing some of its nodes may face a higher latency for the traffic of its own network.

Manuscript received September 19, 2012; revised April 10, 2013, and July 8, 2013; accepted July 18, 2013. The associate editor coordinating the review of this paper and approving it for publication was A. MacKenzie.

G. Quer is with the Qualcomm Institute, University of California San Diego, 9500 Gilman Dr., La Jolla, CA 92093 (e-mail: gquer@ucsd.edu).

F. Librino is with the Italian National Research Council, Via Moruzzi 1, 56124 Pisa, Italy (e-mail: federico.librino@iit.cnr.it).

L. Canzian is with the Dep. of Electrical Engineering, University of California Los Angeles, 420 Westwood Plaza, CA 90095 (e-mail: lcanzian@ucla.edu).

L. Badia, and M. Zorzi are with the Dep. of Information Engineering, University of Padova, Via G. Gradenigo 6/B, 35131 Padova, Italy (e-mail: {badia, zorzi}@dei.unipd.it).

Part of this work has been presented at the IEEE International Conference on Communication (ICC), Ottawa, Canada, Jun. 2012 [1].

This work was partially supported by the U.S. Army Research Office, Grant. No. W911NF-11-1-0538, by the Qualcomm Institute, University of California San Diego, by the CaRiPaRo Foundation, Padova, Italy, under the "Progetti di Eccellenza 2012" program, and by the European Commission under the SAPHYRE project, Grant No. 248001.

Digital Object Identifier XX.XXXX/T-WC.2013.XXXXXX

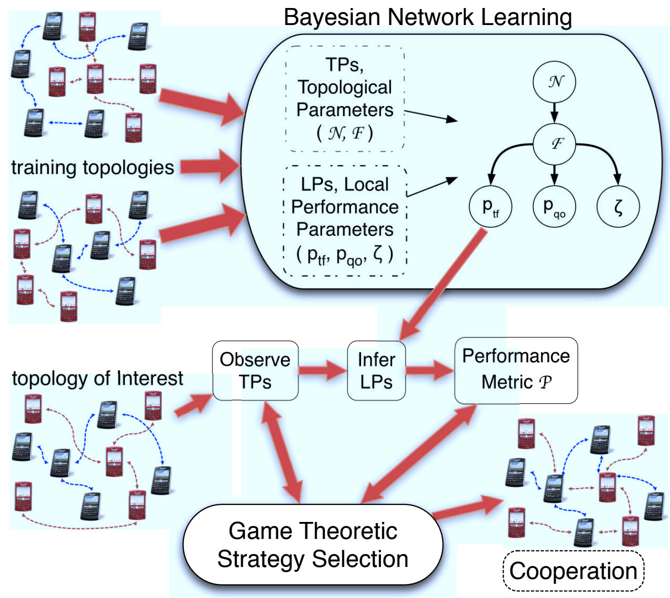

Fig. 1. Logical structure of the proposed approach.

In this paper, we consider two wireless multi–hop networks deployed in the same region but operated by different entities. Each node is sending packets to every other node in the same network. In the case of no cooperation, the two coexisting networks perform their operations separately: each network only uses its own resources to deliver the data packets generated by its nodes. Since they are assumed to share the same spectrum resources, they compete to access the channel, and inter–network interference may limit the overall performance.

In our approach, each network can also share with the other network a limited number of nodes to jointly increase the performance of both networks. The logical structure of the proposed approach is depicted in Fig. 1. During a learning

0000–0000/00$00.00©2013 IEEE

phase, we observe a set of local parameters: some of them are directly observable (i.e., we can assume that each network knows their values), and depend only on the topology of the network (topological parameters, TPs), like the number of neighbors at a given node. Some other parameters depend on the local characteristics of the traffic load (local performance parameters, LPs).

We use the observed data to build the probabilistic relationships among all the parameters, summarized in a Bayesian network (BN). Then we consider the scenario of interest, we observe the TPs in such a scenario, and we use the BN to infer the LPs that will be used to calculate a cost metric. We can finally use this information to model the interaction between the two networks through game theory (GT) and to select the best nodes to be shared in order to minimize the chosen cost metric. We measure through simulation the performance improvement due to cooperation.

Note that we do not need to repeat the learning phase every time the topology changes, since the BN learned from the observation of the training topologies can be reused for every topology of interest. Instead, we need to repeat the GT strategy selection for every new topology. Anyway, given a certain topology, our approach does not affect the run-time operations of the nodes in the networks. Hence, given that the network topology does not change too frequently, our scheme does not impact the average power consumption of the nodes.

In brief, the main contributions of this paper are:

- the definition of the cooperation problem between two networks sharing the same spectrum resources as a strategic game;
- the use of BN theory to learn the probabilistic relationships among a set of parameters in the network, in order to infer the network performance from the observable TPs;
- the definition and analysis of a cooperative game among networks to choose the best nodes to share;
- the implementation of the BN predictor and the strategic game in an actual wireless network simulator that evaluates the network behavior at the physical (PHY), medium access control (MAC) and network layers;
- a performance comparison showing the effectiveness of our algorithm, which achieves the same performance as a fully cooperative approach by sharing only few selected nodes.

The rest of the paper is divided as follows. In Section II we overview some related work. In Section III we describe our network scenario and in Section IV we define the performance metrics that are the basis of the cooperation choices. Then in Section V we overview the BN theory and apply it to infer the network performance as a function of the cooperation choices. In Section VI we detail the game involved among the networks and the cooperation strategy adopted, while in Section VII we present the simulation setup and show the main results. Section VIII concludes the paper.

II. Related Work

In recent years, a huge effort has been put on the investigation of cooperative techniques in wireless networks. Cooperation has been regarded as an effective way of improving the network performance, e.g., in terms of throughput, latency, or energy efficiency. This improvement is granted by letting other terminals help the communication between a wireless source and its intended destination. The study of the cooperation between terminals in a very simple three–node scenario paved the way for a number of cooperation-based protocols. In [3], a TDMA scheme with feedback is adopted, whereas the simultaneous transmission of two cooperating nodes is considered in [4] and [5]. In multi–hop wireless networks, the use of relays can be seen as a form of cooperation, since they create new multi–hop routes. Several protocols have been designed to balance the enhanced link reliability and the increased number of transmissions [6]–[9]. Coded cooperation is developed in [6] and [8], whereas an implementation based on hybrid automatic repeat request (HARQ) is introduced in [9]. The use of relays shows how cooperation can also be exploited for routing purposes, as investigated in [10]–[12]. The choice of the best relay, based on the channel conditions, is discussed in [10], whereas several relays, chosen according to topological criteria, simultaneously cooperate in forwarding a packet in the scheme described in [11]. Finally, a cross-layer approach, where cooperation is exploited in ad hoc networks together with the opportunistic routing paradigm, has been shown in [12].

Although a wide literature is available about cooperation among terminals of the same network, fewer works have focused on cooperation between different networks. In most of them, the idea behind a cooperative behavior of two coexisting networks is to share the spectrum resources. Such a paradigm, known as spectrum sharing, is exploited by primary/secondary cognitive radio networks: an unlicensed network is allowed to exploit the same spectrum assigned to a licensed one, provided that a given QoS is guaranteed to the latter. The spectrum can be shared through strategies exploiting different levels of awareness and coordination, whose performance has been analyzed and discussed in [13] and [14]. In [15], the authors investigated the case where two cellular networks share their own spectrum resources and cooperate in order to minimize the mutual interference, observing a gain inversely proportional to the number of nodes in the networks. Also infrastructure sharing has been considered as a promising cooperation technique for cellular networks; in [16] the sharing of some parts of the network structure is described from a business and regulatory perspective.

To enable the use of cooperation, it is necessary to infer the network gain and cost in advance, thus choosing whether or not it is worth to perform cooperation. Other choices, which require some knowledge about the network, must be made, e.g., which nodes to select as relays. An effective tool to exploit the available information and make a real-time estimate of the expected performance is given by probabilistic graphical models [17]. The use of this probabilistic tool is very promising for wireless network optimization, and has been recently exploited, e.g., in [18] where a BN approach is adopted for predicting the occurrence of congestion in a multi-hop wireless network. The use of Bayesian prediction in a game theoretic framework to allow cooperation is discussed in [19].

In spite of the considerable gain allowed by cooperative transmission, modeling the involved agents as selfish decision-makers usually leads to inefficient non-cooperative outcomes, like in the IEEE 802.11 distributed MAC protocol [20]. In [21], a situation similar to the prisoner's dilemma occurs in a slotted Aloha MAC protocol developed in cooperative, competitive and adversarial scenarios. In this paper, we formulate the problem as a repeated game [22], [23], which consists of a number of repetitions of a base game. In repeated games, users must account for the consequences of their current actions on the evolution of the game, and cooperation is obtained by punishing deviating users in subsequent stages. Repeated interactions have already been applied to the study of cooperative relaying. A packet forwarding mechanism balancing the relaying opportunities that each node gives to and receives from other nodes is proposed in [24]. A virtual currency and a mechanism to charge/reward a player that asks/provides a relay service are introduced in [25] and [26]. Finally, [27] considers a reputation mechanism, where a user gains reputation acting as relay and can choose not to serve users having low reputation.

III. System Model

In this section, we describe the network scenario under investigation from the physical up to the routing layer. In our scenario, two ad hoc wireless networks coexist and share the common spectrum resource. Networks 1 and 2 consist of n_1 and n_2 static terminals, respectively, randomly deployed in the same space. Each node is a source of traffic and generates packets according to a Poisson process with intensity λ packets/s/node. The final destination of each packet is another node in the network chosen uniformly at random. Thus, all the nodes in the network act as both sources and destinations, as well as relays, when needed. Furthermore, time is divided in slots and slot synchronization is assumed across the whole network.

We remark that this is only one exemplifying scenario, and that several details at PHY and MAC layer can be modified without hampering the applicability of the framework. With minor modifications, our approach is particularly suited for multi-hop cellular networks [28], but also for relay-aided cellular networks employing spectrum and/or infrastructure sharing [15], a promising paradigm which is being currently investigated. Vehicular networks, where packets can be sent over multi-hop routes, are another interesting scenario [29].

A. Physical Layer

At the physical layer, code division multiple access (CDMA) with fixed spreading factor is employed to separate simultaneous transmissions, since both networks share the same spectrum resources, and a training sequence for channel estimation is added at the beginning of each transmission. The receiving node, $D^{(j)}$, uses a simple iterative interference cancellation scheme to retrieve the desired packet when M simultaneous communications are received. We define the signal to interference plus noise ratio (SINR) at $D^{(j)}$ for the incoming transmission $T^{(i)}$ from node $D^{(i)}$ as

$$\Gamma^{(i,j)} = \frac{S_f P^{(i,j)}}{N_0 + \sum_{k \neq i} P^{(k,j)}} , \quad (1)$$

where N_0 is the noise power and S_f is the spreading factor. $P^{(k,j)}$ indicates the incoming power at $D^{(j)}$ due to $T^{(k)}$, i.e., for all $k = 1, \ldots, M$:

$$P^{(k,j)} = \frac{P_T |h_{k,j}|^2 d_{k,j}^{-\alpha}}{\chi} , \quad (2)$$

where P_T is the transmission power, which is considered to be the same for all nodes in the network, χ is a fixed path-loss term, $d_{k,j}$ is the distance between the receiving node and the source of $T^{(k)}$, α is the path loss exponent, and $h_{k,j}$ is a complex zero mean and unit variance Gaussian random variable, which represents the effect of multi-path fading. According to (2), each link is statistically symmetric, although the interference levels are likely to be different for $D^{(k)}$ and $D^{(j)}$. According to commonly used channel models, in our scenario we consider a time correlated block fading. Therefore, for the channel between nodes $D^{(k)}$ and $D^{(j)}$, the multi-path fading coefficient in time slot t is

$$h_{k,j}(t) = \rho \ h_{k,j}(t-1) + \sqrt{1-\rho^2}\, \xi , \quad (3)$$

where ρ is the time-correlation factor and ξ is an independent complex Gaussian random variable with zero mean and unit variance, as in [12]. The iterative interference cancellation scheme works as follows:

- the destination node $D^{(j)}$ sorts the M incoming transmissions according to the received SINR, in decreasing order (for simplicity, assume $\Gamma^{(1)} \geq \cdots \geq \Gamma^{(M)}$);
- starting from transmission $T^{(1)}$, $D^{(j)}$ tries to decode the corresponding packet, with a decoding probability that is a function of $\Gamma^{(1)}$ and of the modulation scheme;
- if the packet is correctly received, its contribution is subtracted from the total incoming signal;
- $D^{(j)}$ attempts to decode the transmission with the next highest SINR, $T^{(2)}$, and goes on until the transmission being decoded is the packet of interest.

B. MAC Layer

At the MAC layer, we implement a simple transmission protocol based on a request-to-send/clear-to-send (RTS/CTS) handshake. Every time node $D^{(i)}$ wants to send a packet to node $D^{(j)}$, it checks the destination availability by sending an RTS packet; if $D^{(j)}$ is not busy, it replies with a CTS so that $D^{(i)}$ can start transmitting the packet. Correct reception is acknowledged by means of an ACK packet. In the case of decoding failure, after a random backoff time, node $D^{(i)}$ schedules a new transmission attempt, unless the maximum number of retransmissions M_{tx} has been reached, in which case it discards the packet. Signaling packets only need a single time slot for transmission, and are transmitted twice in order to increase their robustness through time diversity. Data packets may instead span several time slots, and only error detection (with retransmission in case of error) is performed on them.

C. Network Layer

The source and destination nodes are not necessarily within coverage range of each other, so we consider multi-hop transmissions. Two nodes can communicate directly if their distance is less than or equal to the transmission range r. To transmit to destinations that are not within coverage, nodes use static routing tables, which are built using optimized link state routing (OLSR) [30]. Each time a node generates a new packet, or receives a packet to be forwarded, it puts it in the node queue, with first-in-first-out (FIFO) policy. The buffer size b is fixed and equal for all nodes. If a new packet arrives when the buffer is full, it is discarded.

IV. Cost metrics

In this section, we define two different cost metrics that can be used as performance indicators by the networks. These cost metrics can be computed from path parameters (PPs, relative to the source and destination nodes), which in turn can be decomposed in LPs (defined for each single node in the path) that can be estimated through a Bayesian approach.

Given the path from $D^{(i)}$ to $D^{(j)}$, the first PP is the delivery delay $\zeta^{(i,j)}$, defined as the average end-to-end delay of a packet sent along the path, given that the packet is received. The other PP is the packet loss probability $p_{pl}^{(i,j)}$, defined as the probability that a packet is lost along the path. Notice that no end-to-end packet retransmission mechanism is implemented in our network. These PPs are taken into account by each of the cost metrics. In fact, ignoring lost packets (i.e., computing the delay statistics only on correctly delivered packets) may lead to an optimistic evaluation of the network performance under heavy traffic, where few packets actually reach the destination. In this case, a high-loss path might end up being considered better than a more reliable path with a slightly higher delivery delay. The other extreme, i.e., defining the delay contribution of a lost packet as infinite, makes the delay evaluation meaningless. Clearly, neither option is desirable in our case.

In the following we define two cost metrics designed for two different network scenarios. We consider different cost metrics because this paper is not focused on a particular network application, thus, the cost metrics can be thought of as the performance indicators of different scenarios. Moreover, we want to remark that the approach used in this paper does not critically depend on the considered performance metric, and different metrics can be easily accommodated.

A. Lost or not in-time packet rate: $\mathcal{P}_{IT}$

In many applications, the packets should be delivered within a given maximum delay, d_{max}, e.g., in a VoIP application. If a packet successfully reaches the destination after a delay longer than $d_{\max}$, it is considered obsolete and discarded. In this scenario, to calculate a cost metric we should estimate the probability $\hat{p}_{IT}^{(i,j)}$ of in-time delivery of a packet in the path from $D^{(i)}$ to $D^{(j)}$, given that the packet is correctly received. Considering K successful transmissions, each with packet delivery delay $\zeta_k^{(i,j)}$, $k = 1, \ldots, K$, we can estimate

$$\hat{p}_{IT}^{(i,j)} = \frac{\sum_{k=1}^{K} \mathbb{1}\left(\zeta_k^{(i,j)} \leq d_{\max}\right)}{K} , \quad (4)$$

where $\mathbb{1}(\cdot)$ is the indicator function. Thus, the in-time packet arrival rate is

$$\lambda_{IT} = \left(1 - p_{pl}^{(i,j)}\right) \hat{p}_{IT}^{(i,j)} \frac{\lambda}{n-1} , \quad (5)$$

where n is the number of nodes, and the lost or not in-time packet rate can be written as:

$$\mathcal{P}_{IT}^{(i,j)} = \left(p_{pl}^{(i,j)} + \left(1 - p_{pl}^{(i,j)}\right)(1 - \hat{p}_{IT}^{(i,j)})\right) \frac{\lambda}{n-1} . \quad (6)$$

B. Information obsolescence: $\mathcal{P}_{IO}$

In a monitoring application, we assume that each node is tracking a specific signal and we are interested in calculating the average time interval since the last correctly received packet was generated, i.e., the average obsolescence of the information from node $D^{(i)}$ at the receiving node $D^{(j)}$. We recursively define it as:

$$\mathcal{P}_{IO}^{(i,j)} = \left(1 - p_{pl}^{(i,j)}\right)\left(\zeta^{(i,j)} + \frac{\tau}{2}\right) + p_{pl}^{(i,j)}\left(\tau + \mathcal{P}_{IO}^{(i,j)}\right) , \quad (7)$$

where the two terms account for the obsolescence of the information in case of correctly received and lost packets, respectively. In the case of a packet correctly received, we consider that the obsolescence of the last correctly received packet varies linearly from $\zeta^{(i,j)}$ at the moment in which the packet is received, to $\zeta^{(i,j)} + \tau$, immediately before the next packet is received. Thus, the average information obsolescence is given by $\zeta^{(i,j)} + \tau/2$. In the case of a packet loss, an additional time interval τ is added to the information obsolescence every time a packet is lost.[1] From the recursive definition in (7) we obtain:

$$\mathcal{P}_{IO}^{(i,j)} = \tau \frac{p_{pl}^{(i,j)}}{1 - p_{pl}^{(i,j)}} + \zeta^{(i,j)} + \frac{\tau}{2} . \quad (8)$$

Once we have estimated the cost metric for each couple of nodes in the two networks, we can estimate the cost metric of the whole network, $\overline{\mathcal{P}}$, defined as the average of a cost metric (chosen between $\mathcal{P}_{IT}$ and $\mathcal{P}_{IO}$) over all the couples of nodes belonging to the network. The aim of each network is to adopt a cooperation strategy that minimizes its cost metric $\overline{\mathcal{P}}$, as addressed in Section VI. In the following section, we propose a method to decompose $\zeta^{(i,j)}$ and $p_{pl}^{(i,j)}$, needed for the computation of $\overline{\mathcal{P}}$, into local parameters.

C. Computation of the PPs from the LPis

The delivery delay $\zeta^{(i,j)}$ is determined by the number of retransmissions in each link on the path. Indeed, for multi-hop routes, a packet has to wait at each relay node until all

[1] Notice that in our network scenario the packets are received at the destination node in the same order they are transmitted.

the packets ahead in the FIFO queue have been sent. The loss of a packet can be caused either by an excessive number of retransmissions, which lead to a packet drop, or by buffer overflow, i.e., the packet is discarded if the next relay has a full queue. Thus, both the delivery delay $\zeta^{(i,j)}$ and the loss probability $p_{pl}^{(i,j)}$ depend on the channel and interference conditions in each link of the path, that in turn depend on the nodes that the routing protocol selects as relays.

In a static network, it is possible to estimate $\zeta^{(i,j)}$ and $p_{pl}^{(i,j)}$ during a training period, which on the other hand is impractical if the network is dynamic, i.e., in the presence of mobile nodes or time-varying traffic. We propose a different way of estimating the delay and the loss probability, based only on instantaneous topological and routing information. Since a packet sent over a multi-hop path has to traverse a number of nodes before reaching the destination, we decompose the overall path delivery delay and the overall path loss probability into contributions given by the various traversed nodes, and we assume that such contributions are independent. More precisely, the overall delivery delay is given by the sum of the average delays required to traverse every single node (time in queue plus transmission time), whereas the overall loss probability is obtained from the loss probabilities at every node (probability of too many transmission failures and probability of buffer overflow). If $\mathcal{R}^{(i,j)}$ is the set of nodes belonging to the path between $D^{(i)}$ and $D^{(j)}$ (excluding $D^{(i)}$ and $D^{(j)}$), we have:

$$\zeta^{(i,j)} = \zeta^{(i)} + \sum_{h \in \mathcal{R}^{(i,j)}} \zeta^{(h)} , \tag{9}$$

where $\zeta^{(h)}$ is the average time between the arrival of a packet at node $D^{(h)}$ and its reception at the next hop. This delay depends on the next relay; indeed, while the time needed for traversing the queue is the same for all packets, the time required for a successful transmission depends on the channel condition and on the receiver availability, and hence on the next hop chosen. We consider $\zeta^{(h)}$ as averaged over all the packets sent by node $D^{(h)}$ to the next-hop relays.

The packet loss in the multi-hop path is calculated in a similar way, i.e.,

$$p_{pl}^{(i,j)} = 1 - (1 - p_{tf}^{(i)})(1 - p_{qo}^{(j)}) \prod_{h \in \mathcal{R}^{(i,j)}} (1 - p_{tf}^{(h)})(1 - p_{qo}^{(h)}) , \tag{10}$$

where $p_{tf}^{(h)}$ is the probability that a transmission from node h to the next hop fails because the maximum number of retransmissions is reached, and $p_{qo}^{(h)}$ is the probability that a packet correctly received at node $D^{(h)}$ is discarded due to buffer overflow. Furthermore, we notice that $p_{qo}^{(h)}$ depends on the queue of the receiving node $D^{(h)}$, while $p_{tf}^{(h)}$ depends also on which node is used as next hop. For this reason, similarly to $\zeta^{(h)}$, we consider a value averaged over all the neighbors of $D^{(h)}$.[2]

[2]The underlying assumption is that the probabilities $p_{qo}^{(h)}$ and $p_{tf}^{(h)}$, with $h \in \mathcal{R}^{(i,j)}$, are all independent. This is a reasonable assumption since there are multiple flows that contribute to the queue length in each node, and the fading considered is spatially uncorrelated.

V. Bayesian Network estimation of performance parameters

In this section we describe how the values of the LPs are estimated at each node with a BN approach as a function of the TPs. First, we overview the BN method, then we describe the learning phase, in which we collect all the variables in a centralized fashion and exploit this data to learn the BN that describes the probabilistic relationships among them. Finally, we show how the BN is used to estimate the LPs for each node in the network.

A. Bayesian Network preliminaries

A BN is a probabilistic graphical model [17] describing conditional independence relations among a set of M random variables through a directed acyclic graph (DAG), which is composed of vertices and directed edges. A vertex v in the graph represents a random variable, while a directed edge from vertex v to vertex u represents a direct probabilistic relation between the corresponding variables. In this case, we say that node v is a parent of node u.

Learning the DAG is equivalent to calculating an approximate structure of the joint probability distribution among the M variables. In the general case, in which there are no conditional independences among the variables, the joint probability is represented by a complete DAG, in which every pair of nodes is connected by an edge. In this case the joint probability is very complex, so the size of the dataset of realizations needed to learn the quantitative probabilistic relationships can be extremely large. Thus, it is important to study the conditional independences among the variables in order to simplify the joint probability, or, equivalently, to cut some of the edges of the complete DAG.

B. BN: learning phase

The BN learning phase can be summarized in the following algorithm:

1) collect the dataset $\mathcal{D}$ from training topologies;
2) select a set $\mathcal{A} = \{S_1, \ldots, S_{|\mathcal{A}|}\}$ of DAGs (possibly including the best fitting DAG);
3) for each $S \in \mathcal{A}$, calculate θ_S, the maximum-likelihood (ML) parameters of the distribution (parameter learning);
4) choose the DAG S and the parameters θ_S that maximize the score function BIC($S|\mathcal{D}$).

Points 2)–4) are known as the structure learning phase, while point 3), which is part of the structure learning phase, is known as the parameter learning phase of the BN learning algorithm. In the following, we detail how these steps are implemented, and what is the computational complexity of each of them.

Dataset Collection. In this phase, we need to collect a dataset $\mathcal{D}$ with instances of all the parameters involved. $\mathcal{D}$ will be used to learn the probabilistic relationships among the parameters. The TPs are the number of flows $\mathcal{F}^{(i)}$, obtained from the routing table, and the number of neighbors $\mathcal{N}^{(i)}$. These two parameters can be calculated for each node in the network based on the network topology, i.e., with the knowledge of the available relays and of the routing table.

Given a specific training topology, we can evaluate also the values of the LPs $\zeta^{(i)}$, $p_{tf}^{(i)}$, and $p_{qo}^{(i)}$.

In this paper, we simulate several training topologies, as detailed in Section VII, in order to study the probabilistic relationships among all the variables. In a real network, in which the topology changes due to mobility, the instances that populate the dataset $\mathcal{D}$ can be collected during a training phase at the beginning of the communications. Since the nodes are mobile, assuming a sufficiently long training period, the entries collected in this scenario are analogous to the ones collected over several static topologies.

Structure learning. This is a procedure to define the DAG that represents the qualitative relationships between the random variables, i.e., the presence of a direct connection between a pair of variables. We use a score based method [31], i.e., we do not assume any a priori knowledge on the data, but we just analyze the realizations of the variables and we score each possible DAG with the Bayesian information criterion (BIC) [32] that we have chosen as a score function. The BIC is easy to compute and is based on the ML criterion, i.e., how well the data $\mathcal{D}$ is represented by a given structure, and penalizes DAGs with a higher number of edges. The BIC score function can be written as:

$$\mathrm{BIC}(S|\mathcal{D}) = \log_2 P(\mathcal{D}|S, \hat{\theta}_S^*) - \frac{\mathrm{size}(S)}{2} \log_2(N) \ , \quad (11)$$

where S is the DAG to be scored, θ_S^* is the set of parameters of S estimated with ML, and N is the number of realizations for each variable in the dataset. In our case we have $M = 5$ multinomial variables, where each variable v has a finite set of outcomes r_v. We define q_v as the number of configurations over the parents of v in S, i.e., the number of possible combinations of outcomes for the parents of v. A specific realization over all the parents of v is named p. We define also N_{vpk} as the number of outcomes of type k in the dataset $\mathcal{D}$ for the variable v, with parent configuration of type p. N_{vp} is the total number of realizations of the variable v with parent configuration p. Given these definitions, it is possible to rewrite the BIC for multinomial variables as [31]:

$$\mathrm{BIC}(S|\mathcal{D}) = \sum_{v=1}^{M} \sum_{p=1}^{q_v} \sum_{k=1}^{r_v} \log_2\left(\frac{N_{vpk}}{N_{vp}}\right) - \frac{\log_2 N}{2} \sum_{v=1}^{M} q_v(r_v - 1), \quad (12)$$

which reduces the complexity of the scoring function to that of a computationally tractable counting problem. For a detailed comparison of scoring functions for structure learning, please refer to [33]. The best BN structure is then obtained as:

$$S^* = \operatorname*{argmax}_S \mathrm{BIC}(S|\mathcal{D}) \ , \quad (13)$$

and is represented in Fig. 2. We have verified that this BN is the same for all values of λ, while quantitatively the probabilistic relationships, i.e., θ_{S^*}, change with λ.

Parameter learning. This phase consists in estimating the parameters of the joint distribution for each DAG $S \in \mathcal{A}$. To obtain the joint distribution, it suffices to estimate the probability of each variable conditioned by the variables that correspond to its parent nodes in the graph. The parameters of the joint distribution given by the DAG S are θ_S, and the ML choice of the best fitting parameters is:

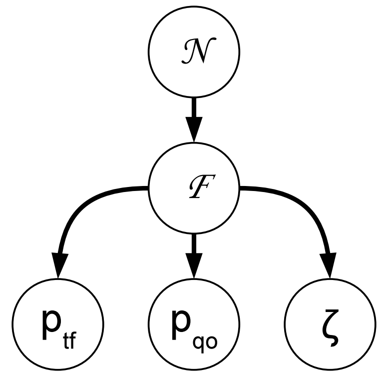

Fig. 2. BN showing the probabilistic relationships among the 5 variables of interest: ζ, p_{tf}, p_{qo} $\mathcal{F}$, and $\mathcal{N}$. This BN is valid for any node $D^{(i)}$ in the network.

$$\theta_S^* = \operatorname*{argmax}_{\theta_S} P(\mathcal{D}|S, \theta_S) \ . \quad (14)$$

In our case this can be seen as a simple counting problem, since all the variables are multinomial, see [17].

Computational complexity The structure learning phase is an NP-complex problem, and in particular the number of structures that should be evaluated with the BIC algorithm increases super exponentially with the number of nodes in the DAG, making the problem intractable even for a moderate number of nodes. In our case, with $M = 5$ variables, there are almost $3 \cdot 10^4$ DAGs, so an extensive search would require to evaluate all the data for each possible DAG to select the DAG that best fits the data according to the BIC criterion. This is computationally very intensive, so we have chosen to perform this search with a hill climbing algorithm, e.g., see [34], that can efficiently find local maxima, and can be restarted multiple times to find a good approximation of the optimal DAG.

For what concerns the parameter learning phase, the ML estimation of the parameters of the distribution is just a counting problem, so each parameter can be estimated in linear time as a function of the size of the dataset, $|\mathcal{D}|$. This phase should be repeated for each choice of $S \in \mathcal{A}$.

We remark that the whole learning phase should be run only at the beginning of the transmissions. The result is a DAG that is valid for each node in the network, can be used to calculate the cost metric for the whole network, and need not be recalculated if the network changes due to mobility or to a different choice of relay nodes.

C. Bayesian Network estimation

The BN estimation algorithm works as follows:

1) for each node in the network, the values of the TPs, $\mathcal{N}$ and $\mathcal{F}$, are observed;
2) the estimation of the LPs is obtained using the joint probability distribution with structure S^* in (13) and parameters $\theta_{S^*}^*$;
3) the expected values of the LPs, from all the nodes, are used to calculate the cost metrics.

After the initial training phase, we can exploit the BN to infer the values of the LPs in constant time, since we just need to apply the joint probability that was previously calculated. Observing the BN structure in Fig. 2, we notice that according to the D-separation rules [17] all the nodes in the DAG are statistically independent if they are conditioned by the observation of the value of $\mathcal{F}$. Thus $\mathcal{N}$ does not influence the values of the three LPs if $\mathcal{F}$ is also observed. Furthermore, once we calculate from the routing table the value of $\mathcal{F}$, we can estimate separately the probability distribution of the three LPs. From these estimated parameters, we can also calculate the cost metric for the network $\overline{\mathcal{P}}$, as detailed in Sec. IV. The expected values of the LPs can be determined by each node, or they can be calculated by a central entity that knows the number of flows $\mathcal{F}$ at each node [3].

We stress the fact that this BN is representative of the general probabilistic relationships among the variables involved, thus the same BN can be exploited for different topologies. In the case of a mobile network, every time the topology changes due to mobility the TPs $\mathcal{N}$ and $\mathcal{F}$ change as a function of the available relay nodes. However, the BN structure remains the same, so we can easily estimate the new LPs.

We should notice that this procedure is different from using a training period to directly derive the LPs in the scenario of interest. Indeed, in that case, a training period would be needed every time the topology changes, in order to estimate the new LPs for each node in the network. On the contrary, with our procedure we can estimate the general joint probability distribution among TPs and LPs, and this probability distribution can be applied to any network topology.

VI. Game theoretic cooperation strategy

Game theory [35] is a branch of applied mathematics that studies strategic situations, called games, in which self-interested and strategic individuals, called players, interact together. The goal is to find equilibria in these games, i.e., a set of strategies from which players are unlikely to deviate.

In this section, we model and analyze through GT the interaction between the two networks. Even though cooperation allows a social gain, it will be actually enabled only by individual choices of all the networks, which individually decide their interests in cooperating, and accurately select the set of nodes to share. We first consider a static game with complete information, which models a one-shot interaction of the networks, and show that the networks do not have any incentive to cooperate in such a context. Then, we consider a repeated game model, which is more suitable for our scenario and allows to trigger cooperation among the networks.

A. Static game with complete information

We label the nodes of the networks from 1 to $n_1 + n_2$, where the nodes in sets $S_1 = \{1, ..., n_1\}$ and $S_2 = \{n_1 + 1, ..., n_1 + n_2\}$ belong to networks 1 and 2, respectively. We formally define the static game with complete information Γ as a tuple $\langle N, \mathcal{A}, U_1, U_2\rangle$, where $N = \{1, 2\}$ is the set of players, i.e., the two networks, and $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ represents the set of action profiles, in which $\mathcal{A}_k$ is the set of actions for player k, $k \in N$. An action $a_k \in \mathcal{A}_k$ represents the set of nodes shared by player k. An operator may not be willing to share too many nodes or some important nodes, e.g., for security or privacy reasons, thus the action set $\mathcal{A}_k$ is a subset of the power set of S_k, $\mathcal{A}_k \subseteq 2^{S_k}$. The utility function $U_k : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ quantifies player k's goodness coming from the adopted actions, and we reasonably assume that this is a decreasing function of $\overline{\mathcal{P}^k}(a_1, a_2)$, which denotes the cost metric for network k, given that the sets of nodes shared are $a_1 \in \mathcal{A}_1$ and $a_2 \in \mathcal{A}_2$. Given the actions a_1 and a_2, the routing tables change accordingly, the number of flows for each node can be computed, $\overline{\mathcal{P}^k}(a_1, a_2)$ can be estimated through the framework introduced in Sections IV and V, and finally the utility $U_k(a_1, a_2)$ can be obtained. In particular, $U_k(\emptyset, \emptyset)$ is the utility of network k when no nodes are shared. We say that an action a_k is non trivial if the shared nodes are exploited by the other network to obtain more efficient paths. Except for the no cooperation action $a_k = \emptyset$, we consider only non trivial actions. Indeed, a trivial action is equivalent to the no cooperation action $\emptyset$. We assume that all the actions are played simultaneously (static property), and all the action sets and utility functions are known by every player (complete information property).

We define the Nash Equilibrium (NE) as the action profile $a^{NE} = (a_1^{NE}, a_2^{NE})$, where each player obtains its maximum utility given the action of the other player, i.e.,

$$\begin{aligned} U_1\left(a^{NE}\right) &\geq U_1\left(a_1, a_2^{NE}\right) \ , \ \forall\, a_1 \in \mathcal{A}_1 \ , \\ U_2\left(a^{NE}\right) &\geq U_2\left(a_1^{NE}, a_2\right) \ , \ \forall\, a_2 \in \mathcal{A}_2 \ . \end{aligned} \tag{15}$$

A NE is an action profile which is stable against unilateral deviations and is an important solution concept in static games with complete information. Unfortunately, the existence and uniqueness of NEs in $\mathcal{A}$ are not guaranteed in general, and NEs can be inefficient from a social point of view. The efficiency is captured by the concept of Pareto optimality. An action profile is Pareto optimal if there exists no other action profile that makes every player at least as well off, while making at least one player better off.

Proposition 1. *The only NE of* Γ *is* $a_1^{NE} = a_2^{NE} = \emptyset$.

Proof: For every action played by the other network, a generic player k strictly prefers playing action $\emptyset$, i.e., not to share any node. In fact, shared nodes strictly increase the traffic handled by the network, which in turn strictly increases the cost metric and strictly decreases the utility. ■

In the considered static game formulation it is not possible to provide incentives for the networks to cooperate, since whatever the other network decides, a network would never want to manage additional flows of packets belonging to the other network. However, we argue that this static formulation is not a proper model for the scenario we have in mind, in which the interaction among the networks is sustained over time. In this case, a repeated game formulation is more suitable.

[3] The number of flows $\mathcal{F}$ at each node can be easily determined knowing the topology of the network and the routing algorithm adopted.

B. Repeated game

We consider the repeated game Γ^R that takes place through stages in time, and at each stage the players play the same stage game Γ repeatedly, knowing the actions the other network has adopted in the previous stages. Because players can condition their play on the information they have received in the past, a player has to take into account how its current action will affect the future evolution of the game.

We define the average utility of a generic network k as

$$U_k^R = (1-\delta) \lim_{T\to+\infty} \sum_{t=1}^{T} \delta^{t-1} U_k^{(t)} \ ,$$

where $U_k^{(t)}$ is the utility obtained by network k at stage t and $\delta \in (0,1)$ is a discount factor, which captures the fact that a present reward is better than a future one.

In the context of repeated games, a player k has to select a strategy $s_k : H_k \to \mathcal{A}_k$ which specifies every action to take conditioned on the past history $h_k \in H_k$. The NE concept defined by (15) for game Γ can be easily extended to the repeated game Γ^R, by substituting the actions with the strategies, the action sets with the strategy sets, and the single stage utility with the average utility. A subgame-perfect equilibrium (SPE) is a particular NE, in which the players' strategies are a NE in every subgame, i.e., starting from any possible stage and for any possible history.

We can design an efficient cooperation strategy profile $s^* = (s_1^*, s_2^*)$, which is a SPE of Γ^R. The key idea is the adoption of a so-called trigger strategy s^* in which the two networks follow by default the cooperation action profile $a^* = (a_1^*, a_2^*)$ and, as soon as one of the two networks deviates from this action profile, the other network punishes it by adopting the no cooperation action $\emptyset$ forever.[4] Being selfish players, the networks select their desired cooperation profile so as to maximize their own gain. Thus, in general, the cooperation profiles preferred by the two networks do not coincide. Inspired by the Nash bargaining solution [35], we assume that the two networks coordinate to the same cooperation action profile a^* solving the following problem[5]

$$a^* = \underset{a\in\mathcal{A}}{\operatorname{argmax}} \left(U_1(a) - U_1(\emptyset,\emptyset)\right)\left(U_2(a) - U_2(\emptyset,\emptyset)\right) \ , \quad (16)$$

subject to:

$$U_k(a) - U_k(\emptyset,\emptyset) > 0 \ , \ k = 1,2 \ .$$

This corresponds to the solution that an impartial arbitrator would recommend to increase in a fair way the utilities of both networks. Notice that such a solution is a Pareto optimal action profile in the stage game Γ. In fact, if there existed an action $a \in \mathcal{A}$ such that $U_k(a) \geq U_k(a^*)$, $k = 1,2$, with the inequality being strict for at least one player, then the constraints would be satisfied also for a and the objective to maximize would be higher in a than in a^*, leading to a contradiction.

Proposition 2. *If (16) has no solution, there exist no cooperation action profile $a^* \neq (\emptyset,\emptyset)$ and trigger strategy s^* such that s^* is a NE of Γ^R.*

Proof: Assume that (16) has no solution; thus, for any trigger strategy s^* that uses a cooperation profile a^*, either network (or both) cannot have a larger utility for a^* than for the no cooperation point. Without loss of generality, assume $U_1(a^*) \leq U_1(\emptyset,\emptyset)$. Thus, if network 2 plays s^*, network 1 can play the no cooperation profile $\emptyset$ achieving on average

$$U_1^R(\emptyset, s_2^*) = (1-\delta)U_1(\emptyset, a_2^*) + (1-\delta)\lim_{T\to\infty}\sum_{t=2}^{T}\delta^{t-1}U_1(\emptyset,\emptyset)$$

$$> (1-\delta)\lim_{T\to\infty}\sum_{t=1}^{T}\delta^{t-1}U_1(\emptyset,\emptyset) = U_1(\emptyset,\emptyset) \geq U_1^R(s_1^*, s_2^*)$$

where the first inequality, i.e., $U_1(\emptyset, a_2^*) > U_1(\emptyset,\emptyset)$, is valid because network 1 can exploit the nodes shared by network 2 to find better paths and only non trivial actions are considered. The last inequality is valid by assumption. Thus, network 1 has an incentive to deviate from s^* and therefore no trigger strategy can be a NE. ■

Since a SPE is a particular NE, if (16) has no solution then no trigger strategy SPE exists. In this case, we assume that the networks never cooperate ($a = (\emptyset,\emptyset)$ is a NE of the stage game Γ, hence it is also a SPE of Γ^R if it is played in every stage). Notice that (16) is without solution if there exists no action profile $a \neq (\emptyset,\emptyset)$ such that both networks can benefit from cooperation, which is possible, though unlikely (it would happen, for example, if the networks were topologically disjoint).

Proposition 3. *If a^* is a solution of (16) and δ is close enough to 1, then the trigger strategy s^* is a SPE of Γ^R.*

Proof: This proposition is a direct consequence of Friedman's theorem [35]. It can be proven by observing that, if either network deviates at step t from the trigger strategy s^*, it gets punished by the other for every subsequent repetition and therefore faces a utility loss from step $t+1$ onwards. Thus, if δ is close enough to 1, the utility loss is sufficiently high to discourage a deviation from the trigger strategy. ■

If the assumptions of Proposition 3 are satisfied, then the two networks can agree to share the nodes according to the cooperation action a^*, and the trigger strategy s^* is a guarantee for both of them that the agreement will be fulfilled.

VII. Results

In this section, we describe the simulation setup and present the performance results of the proposed framework that can be summarized in the following steps: 1) we measure the parameters of interest over several random training topologies with fixed setup; 2) we use the BN method to infer the joint distribution among ζ, p_{tf}, p_{qo}, $\mathcal{F}$, and $\mathcal{N}$, valid for each node in the network; 3) in the scenario of interest, the two

[4] For the sake of simplicity, we avoid considering more complicated strategies, where the cooperation action changes at each stage, as they would cause several intricacies at the network layer, e.g., for what concerns routing table updates and route repairing.

[5] To solve (16) each network must know which are the nodes the other network is willing to share (i.e., the action space) and how its own utility and the utility of the other network are affected by each cooperation possibility. How the two networks collect this information is out of the scope of this paper, and is left for future research.

TABLE I
SIMULATION PARAMETERS

Number of nodes per network	10
Network area size $[m^2]$	200×200
Transmission range, r[m]	75
Transmission power [dBm]	24
Chip rate [chip/s]	7.5×10^6
Noise floor [dBm]	-103
Path loss exponent	4
Path loss fixed term	1000
Fading correlation factor, ρ	0.9
Modulation type	BPSK
Time slot duration [ms]	1
Spreading factor S_f	32
Packet length [bit]	4096
Packet transmission time [slots]	6
Transmission rate, λ [pkts/s/node]	1 to 5
Buffer size b [pkts]	16
Maximum number of MAC retransmissions	5
Initial backoff window [slots]	16
Routing algorithm	OLSR
Simulation duration [slots]	10000

networks evaluate the cost metrics $\overline{\mathcal{P}^k}(a_1, a_2)$, $k \in \{1, 2\}$, for all possible choices of the sets a_1 and a_2 of shared nodes; 4) the two networks compute the two sets of nodes to be shared a_1^* and a_2^* by solving (16), and adopt the trigger strategies s_1^* and s_2^* to enforce the cooperation choice.

A. Simulation Setup

We developed a MATLAB network simulator which encompasses the layers from physical to routing, as described in Section III. We adopt a standard Rayleigh fading channel between each pair of terminals in each time slot, and the BER of each transmission is determined through the BPSK SINR-based expression. The system parameters are reported in Table I, with the packet size being typical of a generic information packet sent in vehicular ad hoc networks. Each simulation run is performed with randomly generated connected networks, deployed on a square area of fixed size, and lasts for 10000 time slots. With the given parameters setup, we first identified, through simulation, the value λ_t of packet generation intensity which results in an end–to–end packet loss probability of 0.1. This can be seen as a threshold value between a lightly loaded and an overloaded network. Different values of the normalized traffic generation intensity $\lambda_n = \lambda/\lambda_t$ were considered, from $\lambda_n = 0.4$ up to $\lambda_n = 2$.

In the learning phase we performed 500 simulation runs (training topologies) for each value of λ_n. The data collected, $\mathcal{D}$, is used for BN learning as detailed in Section V. The joint distribution is derived, and the expected values of the LPs (namely the average delivery delay ζ, the probability of buffer overflow p_{qo} and the probability of transmission failure p_{tf}) conditioned on $\mathcal{F}$ are inferred for every node.

In the subsequent steps, a new set of 500 simulation runs was performed for each value of λ_n. In each run, two networks are again randomly deployed. We investigate the average performance of the networks when 1) no nodes are shared, namely *No Coop*; 2) two randomly chosen nodes are shared, namely *2 Rand*; 3) two nodes selected through the proposed game theoretic approach are shared, namely *2 GT*; 4) all nodes are shared, namely *Full Coop*. To adopt the game theoretic approach, we define the utility function for each network as the reciprocal of the cost metric for that network, i.e., $U_k(a_1, a_2) = [\overline{\mathcal{P}^k}(a_1, a_2)]^{-1}$, and the networks can share either no nodes or exactly 2 nodes. Although our approach can be extended to a number of cooperating nodes larger than 2, our results show that a large fraction of the available cooperation gain is already achieved with this choice. Finally, (16) is solved through an exhaustive search.

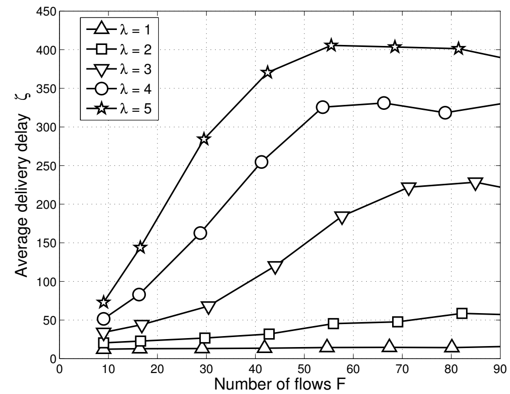

Fig. 3. BN estimation of the average delivery delay ζ as a function of the number of flows $\mathcal{F}$ passing through the node.

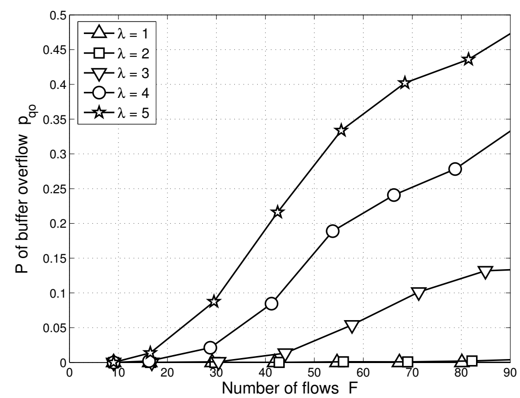

Fig. 4. BN estimation of the probability of buffer overflow p_{qo} as a function of the number of flows $\mathcal{F}$ passing through the node.

B. Bayesian Network estimation

Exploiting the BN approach proposed in Section V, we can evaluate the expected value of the three LPs, as a function of the number of flows $\mathcal{F}$ passing through the node and of the normalized traffic intensity λ_n. The expected values of ζ, p_{qo}, and p_{tf} are shown in Figs. 3, 4 and 5, respectively. Note that there is only a limited number of flows with different sources and destinations that can be injected in the network and, in the considered scenario, this number is very unlikely to exceed

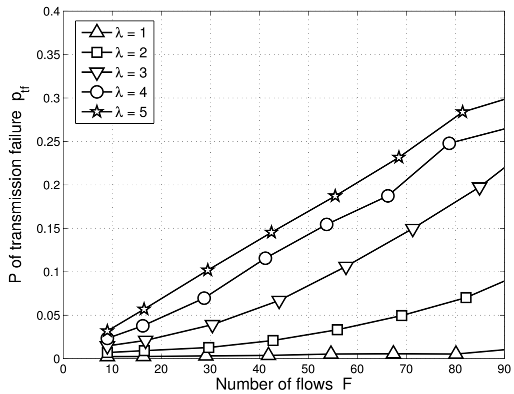

Fig. 5. BN estimation of the probability of transmission failure p_{tf} as a function of the number of flows $\mathcal{F}$ passing through the node.

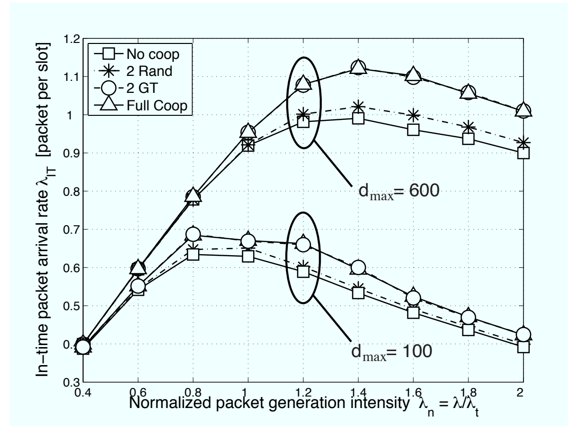

Fig. 7. In-time packet arrival rate λ_{IT} as a function of the normalized packet generation intensity $\lambda_n = \lambda/\lambda_t$ for two values of the maximum allowed delay: $d_{\max} = 100$ and $d_{\max} = 600$, in number of slots.

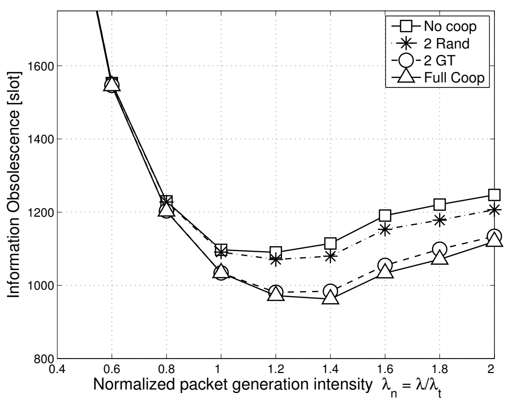

Fig. 6. Information Obsolescence $\mathcal{P}_{IO}$ as a function of the normalized packet generation intensity $\lambda_n = \lambda/\lambda_t$, for the four compared scenarios: with no nodes shared (No Coop); with two nodes shared, randomly chosen (2 Rand); with two nodes shared, chosen via game theory (2 GT); and with all nodes shared (Full Coop).

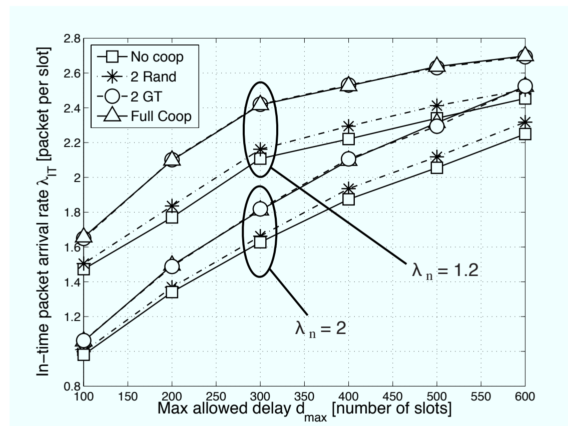

Fig. 8. In-time packet arrival rate λ_{IT} as a function of the maximum allowed delay $d_{\max}$ for two values of the normalized packet generation intensity: $\lambda_n = 1.2$ and for $\lambda_n = 2$.

90, which explains our choice of the scale for the x-axis in the figures. We also observe in Fig. 3 that for very high values of $\mathcal{F}$ and λ_n, the average delivery delay remains stable. We conjecture that this happens since the queues of these nodes are full most of the time, so the time to traverse them cannot grow further.

Notice that the choice of different values for the setup parameters, such as the transmission power or the buffer size, would result in different distributions of the LPs (ζ, p_{qo} and p_{tf}). The framework is therefore able to adapt to different configurations and to derive the corresponding cooperation strategy which best fits the analyzed scenario.

C. Cooperation performance

In Fig. 6, we adopt the cost metric $\mathcal{P}_{IO}$ for the relay selection with the GT approach. We present the actual gain, in terms of reduction of $\mathcal{P}_{IO}$, offered by the considered cooperation strategy. The curves are obtained by averaging over all the random topologies, each consisting of two networks, with $n = 10$ nodes each.

As intuition suggests, full cooperation grants the highest benefits, due to the higher spatial diversity. Hence, this is the minimum achievable cost for the scenario investigated. This benefit is most pronounced when $\lambda_n = 1.4$, i.e., when the cost metric $\mathcal{P}_{IO}$ reaches its minimum as a function of λ_n. When the networks are heavily loaded, cooperation still grants a significant benefit, but the cost $\mathcal{P}_{IO}$ increases, since more packets are lost due to congestion. When only two nodes can be shared, the choice of the shared nodes makes the difference. In fact, Fig. 6 shows that a careful selection of the resources to be shared can significantly increase the achievable gain when compared to a blind random selection. A random selection does not offer any significant gain for lightly loaded networks, while, for heavily loaded networks, it can offer only one third of the gain granted by full cooperation. On the contrary, if the shared nodes are chosen by means of our game-theoretic approach, the maximum achievable gain is almost fully obtained for lightly loaded networks and closely approached for heavily loaded networks.

In Fig. 7 and Fig. 8 we adopt the cost metric $\mathcal{P}_{IT}$ for the relay selection to study the in-time packet performance. Instead of showing $\mathcal{P}_{IT}$, which is decreasing as a function of the normalized packet generation intensity λ_n, we show the performance of the four cooperation strategies in terms of in-time packet arrival rate, λ_{IT}, which has a more interesting behavior, with a global maximum. In Fig. 7 we show λ_{IT} for a maximum allowed delay $d_{\max} = 100$ slots and for $d_{\max} = 600$ slots. We notice that also in this case, an accurate choice of the cooperating nodes made by our cooperation strategy 2 GT achieves the same performance of the case in which all nodes are shared, namely Full Coop. Instead, a random choice of the nodes to share, 2 Rand, provides only a third or less of the total gain achievable with full cooperation.

In Fig. 8, we show λ_{IT} as a function of the maximum allowed delay $d_{\max}$ for a packet generation intensity $\lambda_n = 1.2$ and for $\lambda_n = 2$. We observe that varying the maximum allowed delay $d_{\max}$ with our cooperation strategy 2 GT we obtain the same gain as with full cooperation, while with a random choice of the cooperating nodes we obtain less than a third of the total gain achievable with full cooperation.

We have also studied situations with different numbers of nodes. The results (not reported here due to space constraints) show that the advantage of relay sharing depends on the sizes of the two networks, as expected. In particular, a high node density provides increased diversity to each individual network, thus reducing the benefit of cooperation. An important conclusion is that the proposed framework proved to be able to obtain most of the available cooperation benefit in all cases we considered, reaching a performance very close to that of full cooperation when sharing only a small fraction of the nodes.

D. Extension to a generic number of networks

So far, we have considered for simplicity a scenario with two networks, but the proposed methodology can be applied to a generic number m of coexisting networks. First, the sharing model must be identified. Possible options include the following: 1) each network selects $m-1$ sets of nodes to share, one for each of the other networks; 2) each network selects a single set of nodes to share with all the other networks; and 3) networks are divided in subgroups (e.g., 2 networks for each subgroup) and cooperation is possible only within the same subgroup. 1) represents the most general case, but if there are many networks the best cooperation action may be computationally too expensive, so that in this case 2) and 3) may be preferable. Once the sharing model is identified, the approach described in this paper can be applied: the parameters of interest can be measured over several random training topologies, the BN approach can be used to infer the joint probability distribution among them, a fair cooperation action profile can be computed, and trigger strategies can be used to enforce cooperation. Some minor modifications to the proposed approach are necessary for the design of the trigger strategy. If the computed cooperation action profile is such that any network can benefit from the nodes shared by each of the other networks, then a trigger strategy equivalent to the one considered in the paper can be used (i.e., if network i refuses to share its nodes with network j, then network j punishes it by adopting the no cooperation action forever). If the cooperation gain is more complex (e.g., network i benefits only from the nodes shared by network j, j from the nodes shared by k, and k from the nodes shared by i), then it may be beneficial to consider a trigger strategy in which all other networks jointly punish the deviating one. A full specification of the cooperation strategy and a detailed performance investigation for the case of more than two networks is left for future study.

VIII. Conclusions

In this paper, we have developed a framework which can be used to select the best cooperation strategy between two coexisting wireless networks sharing some of their nodes. A Bayesian network approach has been used to derive the statistical correlation between local topological parameters and global system performance. Based on this information, a game theoretic selection of the nodes which can guarantee the highest benefit has been made. Even when only a small fraction of the nodes is shared, a significant gain can be obtained. In particular, for both lightly and heavily loaded scenarios, the selection scheme based on game theory can achieve almost the same performance as a full cooperation scheme, for the two cost metrics considered.

References

[1] G. Quer, F. Librino, L. Canzian, L. Badia, and M. Zorzi, "Using Game Theory and Bayesian Networks to Optimize Cooperation in Ad-Hoc Wireless Networks," in *Proc. IEEE ICC*, Ottawa, Canada, Jun. 2012.

[2] B. Wang and K. J. R. Liu, "Advances in cognitive radio networks: a survey," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 5–23, Feb. 2011.

[3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[4] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[5] ——, "User cooperation diversity–part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939–1948, Nov. 2003.

[6] T. E. Hunter, S. Sanayei, and A. Nosratinia, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.

[7] P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar, "Cooperative Wireless Communications: A Cross-Layer Approach," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 84–92, Aug. 2006.

[8] Z. Lin, E. Erkip, and A. Stefanov, "Cooperative regions and partner choice in coded cooperative systems," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1323–1334, Jul. 2006.

[9] M. Levorato, S. Tomasin, and M. Zorzi, "Cooperative spatial Multiplexing for Ad Hoc Networks with Hybrid ARQ: Design and Performance Analysis," *IEEE Trans. Commun.*, vol. 56, no. 9, pp. 1545–1555, Sep. 2008.

[10] S. Zhu and K. Leung, "Distributed Cooperative Data Relaying for Diversity in Impulse-Based UWB Ad-Hoc Networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4037–4047, Aug. 2009.

[11] Z. Yang, J. Liu, and A. Host-Madsen, "Cooperative routing and power allocation in ad-hoc networks," in *Proc. IEEE GLOBECOM*, S. Louis, MO, USA, Nov. 2005.

[12] M. Levorato, F. Librino, and M. Zorzi, "Integrated Cooperative Opportunistic Packet Forwarding and Distributed Error Control in MIMO Ad Hoc Networks," *IEEE Trans. Commun.*, vol. 59, no. 8, pp. 2215 – 2227, Aug. 2011.

[13] J. Lee, J. Andrews, and D. Hong, "Spectrum-Sharing Transmission Capacity," *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 3053 – 3063, Sep. 2011.

[14] K. Huang, V. Lau, and Y. Chen, "Spectrum-Sharing between Cellular and Mobile Ad Hoc Networks: Transmission-Capacity Trade-Off," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1256 – 1267, Sep. 2009.

[15] D. Kang, K. Sung, and J. Zander, "Cooperation and Competition between Wireless Networks in Shared Spectrum," in *Proc. IEEE PIMRC*, Toronto, Canada, Sep. 2011.

[16] T. Frisanco, P. Tafertshofer, and R. Ang, "Infrastructure Sharing and Shared Operations for Mobile Network Operators," in *Proc. IEEE ICC*, Beijing, China, May 2008.

[17] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[18] G. Quer, H. Meenakshisundaram, B. Tamma, B. S. Manoj, R. Rao, and M. Zorzi, "Using Bayesian Networks for Cognitive Control of Multi-hop Wireless Networks," in *Proc. IEEE MILCOM*, San Jose, CA, USA, Nov. 2010.

[19] K. Akkarajitsakul, E. Hossain, and D. Niyato, "Distributed resource allocation in wireless networks under uncertainty and application of Bayesian game," *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 120–127, Aug. 2011.

[20] G. Tan and J. Guttag, "The 802.11 MAC protocol leads to inefficient equilibria," in *Proc. IEEE INFOCOM*, Miami, FL, USA, Mar. 2005.

[21] R. Ma, V. Misra, and D. Rubenstein, "Modeling and Analysis of Generalized Slotted-Aloha MAC Protocols in Cooperative, Competitive and Adversarial Environments," in *Proc. IEEE ICDCS*, Lisboa, Portugal, Jul. 2006.

[22] L. Lifeng and H. El Gamal, "The Water-Filling Game in Fading Multiple-Access Channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 722–730, May 2008.

[23] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux, "On Selfish Behavior in CSMA/CA Networks," in *Proc. IEEE INFOCOM*, Miami, FL, USA, 2005.

[24] L. Buttyan and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM J. Mobile Networks*, vol. 8, pp. 579–592, 2003.

[25] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J. Le Boudec, "Self organization in mobile ad hoc networks: the approach of Terminodes," *IEEE Commun. Mag.*, vol. 39, no. 6, p. 166, 2001.

[26] S. Zhong, J. Chen, and Y. Yang, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, 2003.

[27] S. Marti, T. Giulli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proc. IEEE MobiCom*, Boston, MA, USA, 2000.

[28] Y. H. Tam, R. Benkoczi, H. S. Hassanein, and S. G. Akl, "Optimal Channel Assignment in Multi-Hop Cellular Networks," in *Proc. IEEE GLOBECOM*, Washington DC, USA, Nov. 2007.

[29] J. Zhang, Q. Zhang, and W. Jia, "VC-MAC: a cooperative MAC protocol in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1561–1571, Mar. 2009.

[30] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing protocol for ad hoc networks," in *Proc. IEEE INMIC*, Lahore, Pakistan, Dec. 2001.

[31] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*. Springer, 2007.

[32] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[33] S. Yang and K.-C. Chang, "Comparison of Score Metrics for Bayesian Network Learning," *IEEE Trans. Syst., Man, Cybern. A*, vol. 32, pp. 419–428, May 2002.

[34] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[35] G. Owen, *Game Theory*, 3rd ed. New York: Academic, 2001.

Giorgio Quer (S'06, M'12) was born in Asolo, Italy, on July 27th, 1983. He received the B.Sc. degree, the M.Sc. degree (with honors) in Telecommunications Engineering and the Ph.D. degree in Information Engineering from University of Padova, Italy, in 2005, 2007 and 2011, respectively. In 2007 he was a visiting researcher at the Centre for Wireless Communication at University of Oulu, Finland, where he performed research on forwarding and routing techniques for wireless sensor networks. During his PhD studies his research has been focused on wireless networks optimization, as part of the EU-funded SENSEI project and in collaboration with DOCOMO Euro-Labs, Munich, Germany. In 2010 he was a visiting scholar at Calit2, University of California San Diego (UCSD), working on learning techniques for cognitive networks within the military-funded ARO-CogNet project. Since 2011, he is a postdoc researcher at UCSD. He serves as a reviewer for several IEEE and ACM transactions. His research interests include wireless sensor networks, compressive sensing, wireless network optimization, cognitive networking, signal processing, and wireless health.

Federico Librino (S'07, M'10) received an M.Sc. degree (with honors) and a Ph.D. in telecommunication engineering from the University of Padova in 2006 and 2010, respectively. During 2009, he pursued an internship at Qualcomm Inc., San Diego, California. From 2010 to 2012, he was post-doctoral researcher at the University of Padova. Since November 2012, he is at the Informatics and Telematics Institute of the Italian National Research Council, where he is currently working on vehicular networks. His main research interests include cooperation techniques in ad hoc and cellular networks, cross-layer protocol design for wireless networks, cognitive networks modeling.

Luca Canzian (M'13) received the B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from the University of Padova, Italy, in 2005, 2007, and 2013, respectively. From 2007 to 2009 he worked in Venice, Italy, as an R&D Engineer at Tecnomare, a company providing design and engineering services for the oil industry. From September 2011 to March 2012 he was on leave at the University of California, Los Angeles (UCLA). Since January 2013, he has been a PostDoc at the Electrical Engineering Department at UCLA. His research interests include resource allocation, game theory, online learning and real-time stream mining.

Leonardo Badia received his MS (with honors) and PhD from the University of Ferrara, Italy. After having been with the University of Ferrara and the IMT Lucca Institute for Advanced Studies, Italy, in 2011 he joined the University of Padova, Italy, where is currently Assistant Professor. His research interests include protocol design for multihop networks, cross-layer optimization, transmission protocol modeling, and applications of game theory to radio resource management. He serves on the Editorial Boards of the IEEE Transactions on Communications, the IEEE Wireless Communications Letters, and the Wiley Journal of Wireless Communications and Mobile Computing.

Michele Zorzi (S'89, M'95, SM'98, F'07) was born in Venice, Italy, on December 6th, 1966. He received the Laurea and the PhD degrees 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) as a visiting PhD student, working 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, broadband wireless access and underwater acoustic communications and networking.

Dr. Zorzi was the Editor-In-Chief of the IEEE WIRELESS COMMUNICATIONS MAGAZINE from 2003 to 2005 and the Editor-In-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2011, and currently serves on the Editorial Board of the WILEY JOURNAL OF WIRELESS COMMUNICATIONS AND MOBILE COMPUTING. He was also guest editor for special issues in the IEEE PERSONAL COMMUNICATIONS MAGAZINE (Energy Management in Personal Communications Systems) IEEE WIRELESS COMMUNICATIONS MAGAZINE (Cognitive Wireless Networks) and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Multi-media Network Radios, and Underwater Wireless Communications Networks). He served as a Member-at-large of the Board of Governors of the IEEE Communications Society from 2009 to 2011.