On Interference-Aware Cooperation Policies for Wireless Ad Hoc Networks

Cristiano Tapparello, Stefano Tomasin and Michele Rossi
DEI, University of Padova, via Gradenigo 6/B – 35131, Padova, Italy.

***Abstract*—In this paper we devise efficient optimization techniques to find optimal routing and scheduling policies for wireless ad hoc networks in the presence of multi-user interference and cooperative transmissions. Our focus is to assess the impact of interfering among distinct data flows on optimal routing paths and related transmission schedules. In our reference scenario, all nodes have a single antenna and can cooperate in the transmission of packets. Given that, we first model the cooperative transmission problem using linear programming (LP). Thus, for an efficient solution, we reformulate the joint routing and scheduling problem as a *single-pair shortest path problem*, which is solved using the A^* search algorithm through specialized heuristics. Simulation results of the obtained optimal policies confirm the importance of avoiding interfering paths and that interference-aware routing can substantially improve the network performance in terms of throughput and energy consumption, even when the number of interfering paths is small. Our models provide useful performance bounds for the design of distributed cooperative transmission protocols in ad hoc networks.**

I. Introduction

In the past few years wireless networks with interference have been intensively studied, starting from the seminal work by Gupta and Kumar [1]. In [2] it is proven that computing optimal paths considering interference between simultaneous flows is an NP-hard problem. Moreover, [2] points out that one of the key ingredients of efficient routing protocols in the presence of interference is a proper transmission scheduling. Hence, most of the existing literature focuses on the joint optimization of routing and scheduling. [3] provides a multi commodity flow formulation to maximize interference separation, while limiting path inflation (i.e., the average path length). Joint routing and scheduling have been modeled as a network flow problem both ignoring [4] and considering [5] interference among nodes. Also, routing and scheduling models have been combined to route flows with guaranteed bandwidth in [6] and a greedy algorithm has been derived in [7] for their optimization. A similar approach is presented in [8], with a joint optimal design of congestion control, routing and scheduling. While these papers propose viable routing techniques in wireless ad hoc networks with multi-user interference, our focus here is on algorithms that exploit the cooperation among nodes.

Cooperative transmission has been proposed as an effective way of increasing the throughput that, if wisely used, has also the potential of reducing the energy consumption. Early studies dealt with two-hop communication topologies [9] where the transmission between two nodes is assisted by a third node, usually located within them. When multiple hops are considered, cooperative routing becomes relevant and various approaches have been investigated. In [10] it is proposed that a subset of nodes that have received the information at a given hop cooperate in forwarding it to nodes placed farther away. However, the routing path is calculated ignoring cooperation. In [11] the number of cooperative nodes is computed during the network initialization phase. Although sub-optimal, these approaches improve the throughput and reduce energy consumption. Still, further optimizations are possible through the joint optimization of routing paths and transmission schedules.

In this work we combine joint routing and scheduling with node cooperation devising efficient optimization techniques to find optimal transmission policies for ad hoc networks with arbitrary topology. A similar problem has been heuristically addressed in [12], where cooperation policies for multi hop wireless networks with multiple source-destination pairs are studied. According to that scheme, a fixed number of nodes cooperate at each time step. The interference is modeled using contention graphs, where clusters of nodes interfere only if they have nodes in common. Note that this assumption may not hold in practice, as nearby nodes may interfere even though they belong to different clusters. Instead, in this paper we model the interference considering the protocol model introduced in [1].

We hereby consider multi-hop wireless ad hoc networks with a number of concurrent data flows where, for each different flow, nodes decode the message and forward it to the next hop until it reaches the destination. For the transmission cost we consider the weighted sum of the normalized consumed energy and the normalized delay, which we divide by the probability of successful reception accounting for channel impairments, e.g., path loss and fading. Hence, we derive the optimal joint cooperative routing and scheduling policy, determining at each time step and for each flow, which nodes must cooperate to minimize the expected cost over all possible realizations of the data transmission process. To this end, we first model the cooperative routing problem through a linear programming (LP) formulation and subsequently derive an equivalent, but more tractable, *single-pair shortest path problem* [13]. Our results confirm the importance of considering inter-flow interference in the optimization of cooperative transmission policies and provide useful performance bounds for the design of practical protocols.

The rest of the paper is organized as follows. Section II presents the system model, Section III formalizes the joint routing and scheduling problem using linear programming (LP). This LP problem is reformulated as a single-pair shortest path problem in Section IV. Section V presents some numerical results and Section VI concludes the paper.

II. System Model

Consider a wireless network consisting of a set $\mathcal{N}$ of nodes spread out according to any distribution. Time is slotted with a slot corresponding to the fixed transmission time of a packet and all nodes are synchronized at the slot level. The nodes are grouped into clusters during the network initialization phase according to any clustering algorithm. Moreover, only the nodes belonging to the same cluster can cooperate for the transmission of a packet. When multiple nodes cooperate they transmit the same packet simultaneously, i.e., in the same slot. From the original network nodes, we build a superimposed structure of *virtual nodes* on which we perform our optimization. A virtual node can be of three types: T1) a single network node, T2) a cluster of network nodes or T3) a subset of the nodes in a cluster.

We obtain a weighted directed graph $G = (V, E)$, where V is the set of virtual nodes and E is the set of edges, where each edge $(i, j) \in E$ represents a possible communication link between any two given virtual nodes in V. Moreover, each edge $(i, j) \in E$ is weighted with a cost c_{ij} according to a metric that takes into account the energy used for transmission, the reliability of the link and the entangled delay. In G, transmissions and receptions occur between virtual nodes and once a packet is successfully received at a given virtual node, all the actual nodes therein will cooperate for its subsequent transmission in a future slot. In particular, we set

$$c_{ij} = \begin{cases} c & i = j \\ \dfrac{\beta c + (1-\beta) w_i}{p_{ij}} & i \neq j \end{cases}, \tag{1}$$

where c represents a delay cost for the transmission of one packet in the corresponding slot, w_i is the number of actual nodes in the virtual node i and $\beta \in [0, 1]$. Note that $i = j$ means that the packet is not transmitted during a time slot; the virtual node will transmit it in a future slot as dictated by the optimal transmission schedule. In this case, we incur the positive delay cost c so as to avoid unnecessary self-loops during the optimization process, which lead to erroneous solutions. Finally, p_{ij} is the probability that the packet transmitted by virtual node i is successfully received by all nodes in j, as detailed in Section II-A. Note that, considering the use of Stop and Wait ARQ for failed packets, $1/p_{ij}$ is the average number of transmissions for the successful delivery of a packet over link (i, j).[1] Thus, c/p_{ij} and w_i/p_{ij} respectively correspond to the average delay and the average energy expenditure for the successful transmission of the packet over this link.

A demand is a pair of nodes (s, f) with $s, f \in V$ and $s \neq f$ which indicates node s as the source for a packet to be delivered to the final (or destination) node f. The set of demands is denoted by $\mathcal{D} = \{(s_1, f_1), (s_2, f_2), \ldots, (s_K, f_K)\}$. We say that a subgraph $H \subseteq G$ connects a demand (s, f) when it contains a path from s to f, i.e., a sequence of edges $(s, n_1), (n_1, n_2), \ldots, (n_{\ell-1}, n_\ell), (n_\ell, f)$, where each edge corresponds to the transmission in a particular time slot. Note that source s and destination f are virtual nodes of type T1, whereas n_i with $i = 1, 2, \ldots, \ell$ are virtual nodes which, when cooperative transmissions occur, can also be of type T2 and T3. Packet transmissions occur synchronously according to the slotted time structure. Hence, the transmission of a packet through a path that is $\ell + 1$ hops long, with $\ell \geq 0$, entails a minimum of $\ell + 1$ time slots. Note that more than $\ell + 1$ time slots may be needed for the transmission over this path as the packet may stop at some nodes during certain time slots to avoid interference with other flows. Finally, it is assumed that each demand $d \in \mathcal{D}$ is composed of a single information packet.

[1]Neglecting the transmission of acknowledgments.

Given any two nodes $i, j \in \mathcal{N}$, we indicate with $d_{\max}$ the maximum distance at which a packet transmitted from i is received at j with a probability larger than or equal to δ_{th} (with $\delta_{\text{th}} > 0$ and small), or equivalently having an outage probability smaller than or equal to $1 - \delta_{\text{th}}$. In other words, $d_{\max}$ is considered as the maximum distance at which two nodes can reliably communicate. Also, we let $\alpha d_{\max}$ with $\alpha \geq 1$ be the interference range, i.e., the maximum distance for which the transmission from a node i interferes with a concurrent reception at a node j.

To quantify the interference among paths in the presence of cooperative transmissions we need to consider the transmission of virtual nodes. Specifically, we say that two paths interfere with one another in a given time slot when the transmission of one virtual node in the first path interferes with the transmission of another virtual node of the second path. Formally, let $n_i \to n_j$ and $n_h \to n_k$ be the transmissions on the first and second path, respectively, where $n_i, n_j, n_h, n_k \in V$. In this work we consider that $n_i \to n_j$ interferes with $n_h \to n_k$ if either of the following conditions is verified:

C1. There exists at least a pair of nodes with the first being in n_i and the second in n_k with distance smaller than or equal to $\alpha d_{\max}$. In this case the transmission from n_i would interfere with the reception at n_k.

C2. There exists at least a pair of nodes with the first being in n_h and the second in n_j with distance smaller than or equal to $\alpha d_{\max}$. In this case the transmission from n_h would interfere with the reception at n_j.

Following this rationale, we define an *interference graph* $I = (V, A)$, having as vertices the virtual nodes in V. The set A contains the edges and is obtained connecting any two virtual nodes $n_i, n_j \in V$ if there exists at least a pair of nodes with the first being in n_i and the second in n_j with distance smaller than or equal to $\alpha d_{\max}$.

We remark that alternative and more precise conditions for the definition of the interference graph are possible. This definition of interference is adopted here as it is computationally tractable, while providing a reasonable approximation of the actual interference among virtual nodes. Note that these conditions do not impact the correctness of our optimization algorithm, which works for any given interference graph.

A. Calculation of Packet Outage Probabilities

Each node is equipped with one antenna, and when the nodes of a given virtual node $s \in V$ cooperatively transmit, the total number of transmit antennas is w_s, i.e., the number of nodes in s. We assume that nodes operate in half-duplex

$$p_{\text{out}} = 1 - \left(\prod_{j=1}^{N_\sigma} \sigma_j^{-r_j}\right) \sum_{k=1}^{N_\sigma} \sum_{\ell=1}^{r_k} \frac{\phi_{k,\ell}(-\sigma_k^{-1})}{\sigma_k^{-r_k+\ell-1}} f_1\left(\sigma_k^{-1} \frac{2^R-1}{\rho}, r_k - \ell\right) \tag{3}$$

mode and that the same power is used at all transmit antennas. Furthermore, we assume no channel knowledge at the transmitter, i.e., transmit nodes are not aware of position and channel conditions of surrounding nodes. In what follows we compute the outage probability in the presence of path loss and fading when all nodes in s transmit to a single node $j \in \mathcal{N}$.

As transmit nodes are not aware of channel conditions, messages are encoded with a capacity-achieving code having a data rate per unit frequency of R. When the channel capacity, normalized with respect to the bandwidth, is below rate R, outage occurs. In this case the packet is not decoded at the receiving node and is discarded. Let C be the capacity of the channel, normalized with respect to the bandwidth. Then, the outage probability is

$$p_{\text{out}} = \mathrm{P}[C < R] \,. \tag{2}$$

In case of a single antenna per node, the capacity turns out to be the logarithm of a linear combination of central chi square random variables, i.e., $C = \log_2(1 + \rho y)$, where ρ is the average signal to noise ratio (SNR) at the receiving antenna, y is the sum of w_s exponential random variables with means $\Sigma_k = (d_k/d_o)^{-\kappa}$, $k = 1, 2, \ldots, w_s$, where w_s is the number of transmitting nodes (antennas), d_k is the distance between the transmitting and the receiving node, d_o is a constant and κ is the path loss propagation exponent. In the following and without loss of generality we assume $d_o = 1$.

For the general case where some of the means Σ_m are equal, i.e. $\Sigma_k = \Sigma_m$ for some k and m, the outage probability can be obtained using the result in [14]. By letting σ_k, r_k and N_σ be the unique means, their multiplicity and the number of equality classes, respectively, with $k = 1, 2, \ldots, N_\sigma$ and $\sum_{k=1}^{N_\sigma} r_k = w_s$, the outage probability for a receiving node $j \in \mathcal{N}$ when all nodes in $s \in V$ transmit is found as in (3), where $f_1(a, b)$ is the cumulative distribution function of a Poisson variable of parameter a,

$$\phi_{k,\ell}(x) = (-1)^{\ell-1} \sum_{\Omega(N_\sigma,k,\ell)} \prod_j \binom{i_j + r_j - 1}{i_j} \tau_j(x) \,, \tag{4}$$

the set $\Omega(N_\sigma, k, \ell)$ defines partitions of $\ell - 1$ through the positive integer indices i_j, such that $\sum_{j=1,\neq k}^{N_\sigma} i_j = \ell - 1$ and $\tau_j(x) = (\sigma_j^{-1} + x)^{-(r_j+i_j)}$. Simpler expressions for the outage probability hold when all the means are equal or when all the means are different, i.e., $r_k = 1$, $k = 1, 2, \ldots, w_s$, see [15, Section 3.3.1, p. 47] and [16].

III. Joint Optimization of Routing and Scheduling

The goal of this work is to find the *minimum weight set* of non-interfering paths connecting all demands.

For each demand $d \in \mathcal{D}$, let s_d and f_d be its source and destination nodes, respectively. Moreover, for each edge $(i, j) \in E$ let $x_{ij}^d(t)$ be 1 if the packet associated with demand d is transmitted over the link (i, j) in time slot t (transmission $i \to j$ with $i, j \in V$) and $x_{ij}^d(t) = 0$ otherwise. In formulas, our minimum weight set problem can be written as:

$$\min \sum_{d\in\mathcal{D}} \sum_{(i,j)\in E} \sum_{t\geq 0} c_{ij} x_{ij}^d(t) \tag{5a}$$

subject to:

$$\sum_{(j,h)\in E | x_{ij}^d(t-1)=1} x_{jh}^d(t) = 1, \quad d \in \mathcal{D}, \forall\, t \tag{5b}$$

$$x_{ij}^d(t) + x_{lm}^{d'}(t) \leq 1, \quad (l,j) \in A \text{ and } (i,m) \in A \quad d, d' \in \mathcal{D}, d \neq d', \forall\, t \tag{5c}$$

$$\sum_{(i,j)\in E} x_{ij}^d(t_1) = 1, \quad d \in \mathcal{D},\ i = s_d, t_1 \geq 0 \tag{5d}$$

$$\sum_{(j,i)\in E} x_{ji}^d(t_2) = 1, \quad d \in \mathcal{D},\ i = f_d, t_2 \geq 0 \tag{5e}$$

$$x_{ij}^d(t) \in \{0, 1\}, \quad (i,j) \in E,\ d \in \mathcal{D}, \forall\, t \,. \tag{5f}$$

The objective function (5a) corresponds to minimizing the total cost incurred by the transmissions along the paths that connect each demand in $\mathcal{D}$. The constraints are:

- Paths creation: for each demand $d \in \mathcal{D}$ and for any time slot t we have the following two cases: (1) the packet is not transmitted by the current virtual node j, i.e., $x_{jh}^d(t) = 0$ for $h \neq j$ and $x_{jj}^d(t) = 1$ or (2) the packet is transmitted from j to $h \neq j$, i.e., $x_{jh}^d(t) = 1$ for $h \neq j$ and $x_{jj}^d(t) = 0$. (5b) follows as these two cases are mutually exclusive.
- Interference avoidance: for each pair of interfering links and for any time slot t, at most one of the two links can be active (5c).
- Source: for each demand $d \in \mathcal{D}$, there must be a time slot $t_1 \geq 0$ from which the path that connects the demand d starts (5d).
- Destination: for each demand $d \in \mathcal{D}$, there must be a time slot $t_2 \geq t_1$ (this is ensured by condition (5b)) from which the path that connects the demand d ends (5e).
- Link: for each demand and time slot a particular link can only be either active or silent (5f).

The presented optimization problem has a linear objective function and linear constraint functions, thus it can be solved using a linear optimization algorithm [17]. The problem has many variables and constraints so the time and the amount of memory required to find the optimal solution can be extremely large. To deal with these facts we derived an alternative formulation of the problem, which can be solved faster requiring a reduced amount of memory.

IV. Shortest Path Formulation

First of all we introduce the notion of *state*. The system state in the generic time slot t is an ordered K-tuple $\boldsymbol{a}(t) = (a_1, a_2, \ldots, a_K)$, $a_d \in V$ which, for each demand $d \in \mathcal{D}$ represents the virtual node a_d that: 1) has the packet associated with demand d and 2) is allowed to transmit in this slot. A transition from state $\boldsymbol{a} = (a_1, a_2, \ldots, a_K)$ to state $\boldsymbol{b} = (b_1, b_2, \ldots, b_K)$ is possible only if the following two conditions are satisfied: 1) each of the nodes b_i can be reached by a transmission from a_i, i.e., $(a_i, b_i) \in E$ and 2) no interference arises, i.e., $(a_i, b_j) \notin A$, $\forall\ i \neq j$. The cost associated with the transition from state $\boldsymbol{a}$ to state $\boldsymbol{b}$ is calculated using (1) as

$$c(\boldsymbol{a} \to \boldsymbol{b}) = \sum_{i=1}^{k} c_{a_i b_i} \ . \tag{6}$$

Using these definitions, the problem of finding the minimum weighted set of non-interfering paths that connect all demands in $\mathcal{D}$ can be seen as a shortest path problem from the starting state $\boldsymbol{s} = (s_1, s_2, \ldots, s_K)$ to the termination state $\boldsymbol{f} = (f_1, f_2, \ldots, f_K)$. Note that s_i and f_i are all virtual nodes of type T1, i.e., they all correspond to actual network nodes, whereas the intermediate virtual nodes along the path can be of any type. Also, we remark that f_i is the termination sub-state associated with the ith demand, i.e., when the packet of the ith demand arrives at the virtual node f_i this demand is delivered and no further transmissions occur. Given this, the problem is equivalent to the *single-pair shortest path problem* [13] that is studied in graph theory and can be solved, for example, using Dijkstra's algorithm. Due to the large number of states that are generated (the number of states in $\boldsymbol{a}(t)$, i.e., $|V|^K$) it is wise to solve our problem using an adequate algorithm in order to limit the time complexity and the memory space required to solve it. A good choice is the A^* *search algorithm* [18] that speeds up the search using heuristics, whilst returning the optimal policy.

A^* is a best-first graph search algorithm that finds the minimum-cost path on a graph from a given initial vertex $\boldsymbol{s}$ to one final vertex $\boldsymbol{f}$. Since in our case each vertex is a state of our problem we will use the two terms interchangeably. A^* uses a *distance-plus-cost heuristic* function to determine the order in which the search visits the states. For any given state $\boldsymbol{x}$ this function is given by the sum of two functions:

1. The path-cost function: given by the accumulated cost from $\boldsymbol{s}$ to $\boldsymbol{x}$, usually denoted by $g(\boldsymbol{x})$.
2. An admissible heuristic cost: given by an admissible heuristic estimate of the minimum cost from $\boldsymbol{x}$ to $\boldsymbol{f}$, usually denoted by $h(\boldsymbol{x})$.

The term admissible means that $h(\boldsymbol{x})$ must be smaller than or equal to the minimum actual cost from $\boldsymbol{x}$ to $\boldsymbol{f}$, calculated over all possible paths. In our problem, for any given state $\boldsymbol{x}$ we compute the path-cost function $g(\boldsymbol{x})$ as the sum of the costs incurred in the path from $\boldsymbol{s}$ to $\boldsymbol{x}$. Note that this quantity can be accumulated during the search. For $h(\boldsymbol{x})$ we proceed as follows:

1. Given $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$ and the final state $\boldsymbol{f} = (f_1, f_2, \ldots, f_K)$, for each x_i, we compute the minimum cost-path connecting x_i to the corresponding final node f_i. This is accomplished using the Dijkstra's algorithm. Let $h(x_i)$ be the cost of this path.
2. $h(\boldsymbol{x})$ is obtained as $h(\boldsymbol{x}) = \sum_{i=1}^{K} h(x_i)$.

Note that the cost so obtained corresponds to the exact minimum cost when the interference is neglected. Hence, $h(\boldsymbol{x})$ is a lower bound of the cost in the presence of interference and the heuristic is admissible.

As an example of the effectiveness of A^* in reducing the number of states to be visited, we considered a network with 9 clusters of nodes with 3 nodes in each cluster and $K = 3$ demands. For this network we have that $|V| = 63$ and the total number of states is $|V|^K = 250047$. A^* allowed the solution of this problem visiting less than 6000 states in all our results, i.e., less than 2.4% of the total number of states.

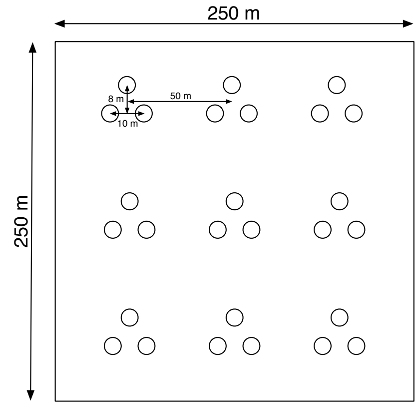

Fig. 1. Network scenario.

V. Numerical Results

In this section we discuss some numerical results obtained using the optimization approach of Section IV on the network topology of Fig. 1. The considered network is composed of 9 clusters of nodes with three nodes per cluster, where clusters are equally spaced in a grid. Therefore from each cluster we obtain 7 virtual nodes. For the following results we picked $c = 1$, $\beta = 0.5$, $\delta_{\text{th}} = 0.1$ (giving $d_{\max} = 58.44$ m) and α is varied from 1 to 2. Thus, we computed the optimal joint routing and scheduling solutions for these settings and we subsequently characterized the performance of these solutions using a simulator. In this simulator, when two links interfere in a given time slot we consider that the corresponding transmissions are lost. In Figs. 2 and 3 we plot the obtained energy and delay performance. For the energy cost we considered the average total number of transmissions carried out in the network for each demand. For the delay we considered the average number of time slots needed to deliver a given demand.

Fig. 2 shows the performance when cooperation is allowed considering two cases: (1) "NoInterf", in this case routing and transmission scheduling policies are obtained neglecting the multi-user interference, i.e., solving separate optimization problems for each demand. The optimal policies for this case are obtained with the algorithm of Section IV setting $\alpha = 0$. (2) "WithInterf", this second case refers to the joint optimal routing and scheduling policies of Section IV. As expected, an increasing number of demands strongly impacts

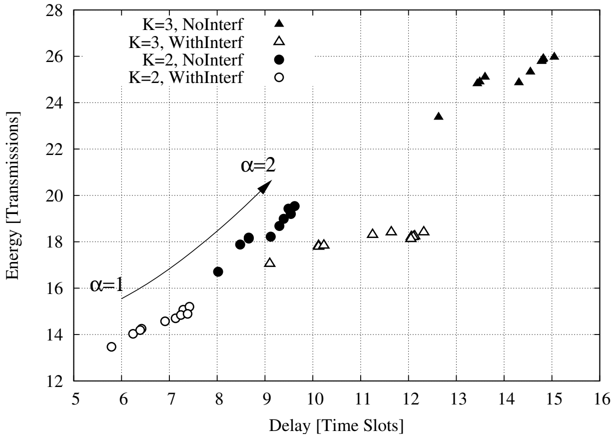

Fig. 2. Energy vs delay varying $\alpha \in [1, 2]$: impact of multi-user interference for $K = 2$ and $K = 3$ demands in the cooperative transmission case.

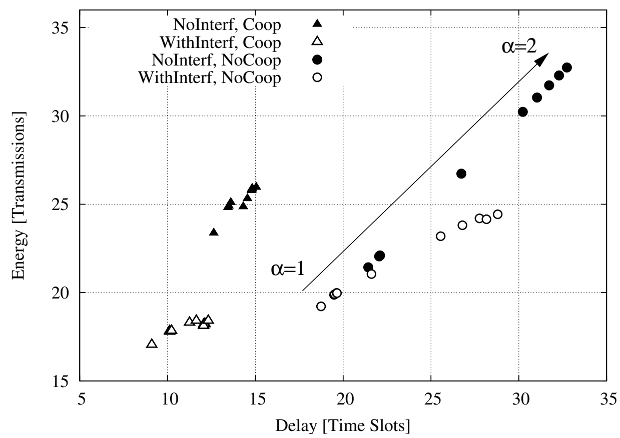

Fig. 3. Energy vs delay varying $\alpha \in [1, 2]$: impact of cooperative transmission for $K = 3$ demands.

the performance, leading to a degradation of energy and delay. However, this performance gap in the case where the interference is neglected is almost doubled for both metrics.

Fig. 3 illustrates the benefits brought about by cooperating transmissions ("Coop" in the figure). First of all, considering the interference in the routing/scheduling policy also leads to better results for both performance metrics and the benefits are substantially larger when nodes cooperate. As expected, the best policies are those accounting for cooperation and interference ("WithInterf, Coop") that, when the interference is high (i.e., $\alpha = 2$), lead to a three fold reduction of both energy and delay. Another interesting observation is that cooperation allows for additional savings in terms of energy and delay with respect to "WithInterf, NoCoop", where the interference is considered but the cooperation is neglected.

In non cooperative systems the interference can be neglected when interference and transmission ranges are equal (see Fig. 3 with $\alpha = 1$). This is no longer valid when nodes cooperate. In fact, in this case, even when $\alpha = 1$ the actual transmission range of multiple cooperating nodes becomes higher than that of a single transmitting node, leading to a larger interference area and thus exacerbating the negative effect of interfering transmissions. On the other hand, when interference is considered in the optimization process, cooperation can provide further benefits of up to 25% and 58% for the energy and delay, respectively (see "WithInterf, NoCoop" versus "WithInterf, Coop" in Fig. 3).

VI. Conclusions

In this paper we solved the joint routing and transmission scheduling problem in wireless ad hoc networks in the presence of multi-user interference and cooperative transmissions. The problem has been formulated using linear programming and, for the sake of an efficient implementation, subsequently solved through a shortest path optimization method exploiting the A^* heuristic search [18]. Numerical results show that cooperative transmissions can respectively provide benefits of up to 25% and 58% for the energy and delay with respect to a non-cooperative approach. The obtained results are useful performance bounds for the design of practical cooperation schemes, which are the objective of our future research.

References

[1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[2] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless Networks*, vol. 11, no. 4, pp. 471–487, July 2005.

[3] V. Kolar and N. B. Abu-Ghazaleh, "A Multi-Commodity Flow Approach for Globally Aware Routing in Multi-Hop Wireless Networks," in *IEEE PERCOM*, Pisa, Italy, Mar. 2006.

[4] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem," in *ACM MobiCom*, San Diego, CA, USA, Sept. 2003.

[5] ——, "The effect of interference on the capacity of multihop wireless networks," in *IEEE ISIT*, Chicago, IL, USA, June 2004.

[6] A. Capone and F. Martignon, "A Multi-Commodity Flow Model for Optimal Routing in Wireless MESH Networks," *Journal of Networks*, vol. 2, no. 3, pp. 1–5, June 2007.

[7] Z. Wu and R. Dipankar, "Integrated routing and MAC scheduling for single-channel wireless mesh networks," in *IEEE WOWMOM*, Newport Beach, CA, USA, June 2008.

[8] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," in *IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[9] T. Cover and A. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.

[10] A. E. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative Routing in Static Wireless Networks," *IEEE Transactions on Communications*, vol. 55, no. 11, pp. 2185–2192, Nov. 2007.

[11] S. Lakshmanan and R. Sivakumar, "Diversity Routing for Multi-hop Wireless Networks with Cooperative Transmissions," in *IEEE SECON*, Rome, Italy, June 2009.

[12] J. Zhang and Q. Zhang, "Cooperative Routing in Multi-Source Multi-Destination Multi-Hop Wireless Networks," in *IEEE INFOCOM*, Phoenix, AZ, USA, Apr. 2008.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 2nd ed. MIT Press, Sept. 2001.

[14] S. V. Amari and R. B. Misra, "Closed-form expressions for distribution of sum of exponential random variables," *IEEE Transactions on Reliability*, vol. 46, no. 4, pp. 519–522, Dec. 1997.

[15] P. V. Mieghem, *Performance Analysis of Communications Networks and Systems*, 1st ed. Cambridge University Press, Apr. 2006.

[16] M. Biswal, N. Biswal, and D. Li, "Probabilistic linear programming problems with exponential random variables: A technical note," *European Journal of Operational Research*, vol. 111, no. 3, pp. 589–597, Dec. 1998.

[17] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, Feb. 1997.

[18] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, Dec. 2002.