

The Impact of Interference on Optimal Multi-path Routing in Ad Hoc Networks

Roland de Haan, Richard J. Boucherie, and Jan-Kees van Ommeren

Stochastic Operations Research, University of Twente, The Netherlands

Abstract. We develop a queueing model characterizing explicitly the impact of interference on end-to-end performance measures such as throughput in ad hoc networks, emphasizing the performance trade-off between single-path and multi-path routing. It may seem attractive to employ multi-path routing, but as all nodes share a single channel, efficiency may drop due to increased interference levels thus yielding single-path performance for some topologies. We formulate a nonlinear programming problem to optimize network performance. Next, we focus on network capacity and show that for this objective the optimum could be found by solving an exponential number of linear programmes. We propose a greedy algorithm that efficiently searches these programmes to approximate the optimal solution. Numerical results for small topologies provide structural insight in optimal path selection and demonstrate the excellent performance of the proposed algorithm. Besides, larger networks and more advanced scenarios with multiple source-destination pairs and different radio ranges are analyzed.

Keywords: Ad hoc networks, Interference, Capacity, Multi-path routing, Network optimization.

1 Introduction

Ad hoc networks have received considerable attention in the recent literature. The main focus has been on the development of routing protocols, since protocols for wired networks cannot be employed efficiently in a wireless let alone an ad hoc environment. The main body of routing protocol proposals regards single-path routing, i.e., for each source-destination pair a single (shortest) path is discovered and used for data transmission (see e.g., [1,2,3]). An alternative is multi-path routing (see [4]) in which multiple paths are used, thereby offering more opportunities for regulating the traffic over the network.

Multi-path routing enhances single-path routing mainly in two directions: (i) to have backup paths available in case of path failures (see e.g., [5,6]), and (ii) to spread traffic to increase the effective bandwidth (see e.g. [7,8]). However, there are also a number of drawbacks to employing multiple paths. In single-path routing the shortest path is normally selected; hence, any additional path will typically be longer. This may not matter when considering capacity questions, but it definitely does when considering the transfer time of a packet. Another,

frequently underexposed, drawback, which occurs typically in wireless environments, is that nodes situated on nearby paths may interfere. As a result, resources need to be shared (to prevent packet collisions) and throughput may decrease drastically. Therefore, the actual performance gain (in terms of bandwidth) of using multiple paths over using a single path is uncertain.

Recently, this bandwidth gain has received more attention, with the emphasis on the development of routing protocols. These protocols often aim at finding link- or node-independent paths and do not explicitly take signal interference between paths into account (see e.g. [9]). An explanation for this is perhaps that the notion of interference is hard to quantify. In fact, it is still an open question how one can accurately measure the level of interference in a network, despite the several metrics that have been suggested (see e.g., [10,11]).

Following a graph-theoretical approach for a given (finite) network instance, [12] and [13] analytically assess the network capacity by taking interference into account. In [12], a multi-commodity flow problem is formulated and extended by interference-related constraints in order to find lower and upper bounds for the capacity. These additional constraints follow by regarding cliques and independent sets in the so-called conflict graph of the network. The disadvantages of this approach are that a central scheduling entity is assumed and that extensive computations are required, even for small networks. The approach of the authors in [13] is similar but aims at a more distributed way of controlling the traffic streams in the network. They develop a low-complexity algorithm to find approximate cliques and use this to calculate lower and upper bounds for the capacity.

In this paper, we study optimization of network performance while explicitly considering the interference between nodes (and thus also paths). More specifically, the trade-off between the bandwidth gain using multiple paths on the one hand and the loss of bandwidth due to the additional interference involved when using those paths on the one hand is investigated. In contrast to most of the work that has appeared in this area, which focuses primarily on the capacity value itself, our interest is mainly in the underlying fundamental aspects so as to gain structural insights into the network performance under interference. First, we discuss the ad hoc network model and then present a general stochastic framework in which many performance metrics of interest can be investigated. The analysis is partly based on the MAC layer IEEE 802.11 protocol for which it has been shown that single-hop ad hoc networks can successfully be modelled by a Processor Sharing queue [14]. We formulate a nonlinear programming problem to optimize network performance. This program is then customized to meet the objective of maximal network capacity as might be achieved by a central scheduler. To solve the resulting quadratic programming problem, an approximate greedy algorithm is proposed. Finally, example topologies are analyzed both to assess the quality of the algorithm and to uncover structural relations in the network. These insights may be applied in the development of routing protocols. However, the design of such protocols is outside the scope of this paper.

The paper is organized as follows. Section 2 describes the ad hoc network model, our mathematical framework and the network optimization problem.

Section 3 presents techniques towards solving the optimization problem, and Sect. 4 discusses the impact of interference on network capacity via illustrative example topologies. A discussion and conclusions are presented in Sect. 5.

2 Model

2.1 Ad Hoc Network Model

Consider an ad hoc network in which all stations (or nodes) are equipped with an identical packet radio (with omnidirectional antenna) operating in half-duplex mode, and transmit over a common channel at identical (maximum) power. A node may transmit data to nodes that are within its *transmission range*. During data reception at a node, all nodes within its *interference range* must be silent for the reception to be successful. Typically, the interference range exceeds the transmission range. Links (or connections) between nodes are assumed error free. However, link errors may easily be incorporated in our model as will be discussed in Sect. 5. We assume a distributed transmission scheduling mechanism that mimics the IEEE 802.11 MAC protocol, which aims to preventing packet collisions. Data transmission in the network is between source-destination pairs (SD-pairs).

2.2 Mathematical Framework

We consider a network consisting of a set of nodes $\mathcal{N} = \{1, \dots, N\}$. This set comprises a collection of source nodes $\mathcal{S} = \{s_1, \dots, s_F\}$ and destination nodes $\mathcal{D} = \{d_1, \dots, d_F\}$, where F denotes the total number of SD-pairs. The remaining nodes can be seen as (pure) relay nodes, but we note that source and destination nodes can also relay traffic. For $j \in \mathcal{N}$, let $\mathcal{N}_T(j) \subset \mathcal{N}$ denote the transmission neighborhood of node j , that is, the set of nodes (possibly sources or destinations) that node j can successfully transmit packets to, and let $\mathcal{N}_I(j) \subset \mathcal{N}$ denote the interference neighborhood of node j , that is, the set of nodes that must be quiet for successful packet reception at node j . The neighborhood relation is not necessarily a symmetrical relation, e.g., $n \in \mathcal{N}_T(n')$ does not imply $n' \in \mathcal{N}_T(n)$. When $n' \in \mathcal{N}_T(n)$, we say that the network contains a *link* from node n to node n' . Paths in the network consists of a number of links starting at a source node and ending at a destination node. On a path from source s to destination d consisting of $\ell + 2$ links via nodes n_1, \dots, n_ℓ , the nodes must be such that $n_1 \in \mathcal{N}_T(s)$, $n_j \in \mathcal{N}_T(n_{j-1})$, $j = 2, \dots, \ell$, $d \in \mathcal{N}_T(n_\ell)$.

Source s_f generates data flows according to a Poisson process at rate $\lambda^{(f)}$ for SD-pair f . A flow consists of a series of data packets. Let $\beta^{(f)}$ denote the mean number of packets per flow between SD-pair f . Then $\alpha_j^{(f)} = \lambda^{(f)} \beta^{(f)}$, for $j = s_f$, is the mean rate at which source s_f generates packets for SD-pair f ; moreover, $\alpha_j^{(f)} = 0$, $\forall j \in \mathcal{N}: j \neq s_f, d_f$, and we set $\alpha_j^{(f)} = -\lambda^{(f)} \beta^{(f)}$, for $j = d_f$. We assume that ordering of packets at the destination can be handled without loss of information. Thus, a flow may be split and its packets may be transferred over different paths selected according to a suitable network optimization mechanism.

Let node j forward a fraction $p_{jk}^{(f)}$ of its incoming packets for SD-pair f to node k . Denote by $\rho_{t,j}^{(f)}$ and $\rho_{r,j}^{(f)}$ the average number of packets transmitted and received, respectively, per time unit by node j for SD-pair f . It is assumed here that for any node transmitting a packet takes exactly one unit of time. We may thus also interpret $\rho_{t,j}^{(f)}$ as the average fraction of time node j is transmitting packets for SD-pair f . Then, for the network to sustain all these packet transmissions, the following relations must hold:

$$\rho_{t,j}^{(f)} = \alpha_j^{(f)} + \rho_{r,j}^{(f)}, \quad \forall j \in \mathcal{N}, \quad \forall f \in \mathcal{F}, \quad (1)$$

$$\rho_{r,j}^{(f)} = \sum_{k \in \mathcal{N}} \rho_{t,k}^{(f)} p_{kj}^{(f)}, \quad \forall j \in \mathcal{N}, \quad \forall f \in \mathcal{F}. \quad (2)$$

Further, we define for node j , $\rho_{t,j} := \sum_f \rho_{t,j}^{(f)}$ and $\rho_{r,j} := \sum_f \rho_{r,j}^{(f)}$. Notice that for nodes j which are pure relay nodes $\rho_{t,j}^{(f)} = \rho_{r,j}^{(f)}$, $\forall f \in \mathcal{F}$. Optimal network design then corresponds to optimal selection of the fractions $p_{kj}^{(f)}$ in (2), so as to maximize a network performance criterion. These fractions determine the optimal path selection for flows in the network.

The number of packets arriving at a node determines the workload of the node. As a consequence, from the flow perspective, flows share the transmission capacity of the nodes. A single node can handle packets originating from different flows and these packets will be transferred in order of arrival. As nodes are identical, each node transmits packets at a normalized unit rate in the absence of signal interference. Clearly, interference among neighboring nodes reduces the amount of time a node is allowed to transmit packets. In a coordinated network, when neighboring nodes each have packets to transmit, these nodes will share the resources. This is achieved, for example, by the MAC layer protocol of IEEE 802.11, where each node uses its fair share of the medium. It is demonstrated in [14] that the PS queue is an adequate model for MAC layer sharing among multiple nodes. The overhead of the MAC layer protocol results in a reduced data rate: for IEEE 802.11 under RTS/CTS the MAC layer operates at roughly 85% efficiency. As a consequence, we may model all nodes in the interference neighborhood (which include the nodes that transmit packets to j) of node j as a single PS queue. We normalize this maximum transmission rate of the nodes at 1. The resulting interference restriction under which node j can still receive data successfully is

$$\sum_{m \in \mathcal{N}_I(j)} \rho_{t,m} + \rho_{t,j} \leq 1, \quad \forall j \in \mathcal{N}. \quad (3)$$

Notice that this restriction is conservative: when none of the multiple transmissions overheard at a node are directed to this node, then those are unnecessarily prohibited. Moreover, the capacity restriction need not be imposed when node j is not a recipient of any data in the optimal design. This is incorporated via the following modification:

$$\rho_{r,j} \left(\sum_{m \in \mathcal{N}_I(j)} \rho_{t,m} + \rho_{t,j} \right) \leq \rho_{r,j}, \quad \forall j \in \mathcal{N}. \quad (4)$$

Our aim is to investigate the performance trade-off between single-path and multi-path routing. In particular, we investigate the maximum data rate (i.e., capacity) that can be sustained by this optimal path selection under interference. Therefore, we consider capacity optimization of the network in equilibrium. However, our framework also allows us to consider alternative performance measures such as the maximal delay for a given traffic load.

2.3 Network Optimization Formulation

Optimal design of paths in the ad hoc network comes down to determining the $p_{jk}^{(f)}$ as the fraction of SD-flow f routed from node j to node k , or equivalently as the probability of routing traffic of SD-flow f from node j to node k . Define the matrices $\boldsymbol{\rho}_t = (\rho_{t,j}^{(f)})_{j \in \mathcal{N}, f \in \mathcal{F}}$ and $\mathbf{p} = (p_{jk}^{(f)})_{j,k \in \mathcal{N}, f \in \mathcal{F}}$. Network optimization can then be formulated as a nonlinear programming problem:

$$\max \quad h(\boldsymbol{\rho}_t, \mathbf{p}) \quad (5)$$

$$\text{s.t.} \quad \rho_{t,j}^{(f)} - \rho_{r,j}^{(f)} = \alpha_j^{(f)}, \quad \forall j \in \mathcal{N}, \quad \forall f \in \mathcal{F}, \quad (6)$$

$$\rho_{r,j}^{(f)} - \sum_{k: j \in \mathcal{N}_T(k)} \rho_{t,k}^{(f)} p_{kj}^{(f)} = 0, \quad \forall j \in \mathcal{N}, \quad \forall f \in \mathcal{F}, \quad (7)$$

$$\rho_{r,j} \cdot (\sum_{m \in \mathcal{N}_I(j)} \rho_{t,m} + \rho_{t,j}) - \rho_{r,j} \leq 0, \quad \forall j \in \mathcal{N}, \quad (8)$$

$$1 - \sum_{k \in \mathcal{N}_T(j)} p_{jk}^{(f)} = 0, \quad \forall j \in \mathcal{N}: j \neq d_f, \quad \forall f \in \mathcal{F}, \quad (9)$$

$$\rho_{t,j} - \sum_f \rho_{t,j}^{(f)} = 0, \quad \forall j \in \mathcal{N}, \quad (10)$$

$$\rho_{r,j} - \sum_f \rho_{r,j}^{(f)} = 0, \quad \forall j \in \mathcal{N}, \quad (11)$$

$$\rho_{t,j}^{(f)}, \rho_{r,j}^{(f)} \geq 0, \quad \forall j \in \mathcal{N}, \quad \forall f \in \mathcal{F}, \quad (12)$$

$$p_{jk}^{(f)} \geq 0, \quad \forall j, k \in \mathcal{N}, \quad \forall f \in \mathcal{F}, \quad (13)$$

where $\mathbf{1}$ is the indicator function and $h(\boldsymbol{\rho}_t, \mathbf{p})$ is a general objective function. For instance, for $h(\boldsymbol{\rho}_t, \mathbf{p}) = \sum_{i=1}^F w_i \cdot \rho_{t,s_i}^{(i)}$ with weights $w_i \geq 0$, $i = 1, \dots, F$, the total weighted capacity in the network is optimized, and for $h(\boldsymbol{\rho}_t, \mathbf{p}) = -\sum_{j=1}^N \frac{\rho_{t,j}}{1-\rho_{t,j}}$ the mean number of packets in the network can be minimized for a given set of flows a_1, \dots, a_F . The first set of constraints (6)–(7) refers to the traffic equations (1) and (2), and describes flow conservation; Eq. (8) is the interference constraint; Eq. (9) indicates that the fractions $p_{jk}^{(f)}$ must sum to one.

The feasible region (6)–(13) shows that we are dealing with a nonlinear programming problem in the unknowns $\rho_{t,j}^{(f)}$ and $p_{jk}^{(f)}$. Our interference assumptions yield constraints that are quadratic in $\rho_{t,j}^{(f)}$ (see Eqs. (7), (8) and (11)) and moreover interacting terms of $p_{jk}^{(f)}$ and $\rho_{t,j}^{(f)}$ show up in Eq. (7). These latter interactions terms can, however, conveniently be eliminated by introducing new variables referring to *link* flows. To this end, introduce $\lambda_{jk}^{(f)} := \rho_{t,j}^{(f)} \cdot p_{jk}^{(f)}$, $\forall j, k \in \mathcal{N}$, $\forall f \in \mathcal{F}$, which represent the amount of traffic going from node

j to node k per time unit for SD-pair f . The constraints (6)–(8) then become linear in $\lambda_{jk}^{(f)}$.

In the rest of this paper, we will focus on the linear objective of capacity optimization. We note that this optimization problem for a single SD-pair resembles the well-known max-flow problem in discrete optimization. For multiple SD-pairs, it resembles the multi-commodity flow problem [15]. Unfortunately, due to the nonlinear interference constraints, our optimization problem cannot be recast in the framework of these problems.

3 Solution Techniques

We approach the program of (5) by exact and approximative solution techniques which are based on the following important observation. Let the problem be formulated in the link flows $\lambda_{jk}^{(f)}$ ($= \rho_{t,j}^{(f)} \cdot p_{jk}^{(f)}$), so that the program of (5) becomes linear, except for the interference constraints. We observe that these quadratic constraints can be replaced by linear ones if the nodes which do not receive any data packets in the optimal traffic distribution (i.e., the nodes with $\rho_{r,j} = 0$) are known. More explicitly, when $\rho_{r,j} = 0$, the constraint is always satisfied, and when $\rho_{r,j} > 0$, we can divide by $\rho_{r,j}$ to get the equivalent constraint:

$$\sum_{m \in \mathcal{N}_I(j)} \rho_{t,m} + \rho_{t,j} \leq 1, \quad \forall j \in \mathcal{N}. \quad (14)$$

3.1 Exact Approach

The above observation shows that any feasible solution of the nonlinear programming problem characterizes a linear programming problem. The global optimum is also a feasible solution and, therefore, we could solve our nonlinear programming problem by consecutively solving 2^N linear problems. However, in practice such an approach is not computationally feasible.

Common techniques would then define the interference constraints by means of functions which indicate whether a node receives data or not. Instead of Eq. (8), we can write:

$$\begin{aligned} \sum_{m \in \mathcal{N}_I(j)} \rho_{t,m} + \rho_{t,j} &\leq 1 + N(1 - r_j), \quad \forall j \in \mathcal{N}, \\ \rho_{r,j} &\leq r_j, \quad \forall j \in \mathcal{N}, \\ r_j &\in \{0, 1\}, \quad \forall j \in \mathcal{N}. \end{aligned} \quad (15)$$

The introduction of the indicator function r_j transforms the quadratic programming problem into a mixed integer programming problem that is linear in the $\lambda_{jk}^{(f)}$. Although such a mixed problem is NP-hard, the advantage of this formulation is that standard solvers for this class of programming problems are widely available. Such standard solvers often embed branch-and-bound techniques to reduce the number of LP problems to be solved. Although such branch-and-bound techniques provide the optimal solution, unfortunately no guarantees as to the number of programs to be solved can be given.

3.2 Greedy Approximation Approach

For the evaluation of large networks, a more efficient technique than inspecting all LP problems or applying a branch-and-bound technique will be required. To this end, we introduce an approximate greedy algorithm which works linearly in N , the number of nodes. Our greedy algorithm is defined as follows.

Initially, assume that all nodes in the network receive data (i.e., $\rho_{r,j} > 0, \forall j \in \mathcal{N}$) and then solve the linear program (i.e., with all interference constraints included) and its corresponding dual. In each following step, a node j^* is eliminated from the network and the resulting (linear) program is then analyzed again. That means, the program with ρ_{r,j^*} set to zero and thus with one interference constraint removed. This iteration process is continued until the optimum decreases after a node elimination.

The key element of the algorithm is the elimination step. This elimination takes place based on the values of the dual variables related to the interference constraints, since these dual values indicate the importance of the primal constraints (which have a one-to-one correspondence to the nodes). Therefore, in each step we eliminate the node with the greatest dual value (which is strictly positive for a connected network), which means that its constraint is definitely tight by appealing to the complementary slackness conditions. Notice that removing a node corresponding to a primal constraint that is not tight (i.e., its dual value is zero) would never lead to an improvement of the objective function. In the case of a tie, i.e., multiple nodes attain the highest dual value, then we select one of the nodes (by solving the primal problem) with the lowest fraction of transmitted packets per unit time. The rationale behind this is that a limited fraction can more easily be diverted via other paths.

The great advantage of this greedy approach is that since the set of removable nodes is of size N , if the greedy algorithm is used only order N of the 2^N LP problems have to be solved.

4 Numerical Results

We evaluate the impact of interference on the network capacity for various scenarios. We discuss a number of basic topologies that provide structural insights. Next, we move to more general topologies to assess the performance of our proposed greedy algorithm. Finally, capacity results for multiple SD-pairs and for extended interference and transmission ranges will be discussed. Recall that the network capacity is defined in terms of the fractions of time that the sources are transmitting and is thus a dimensionless metric.

4.1 Single Source-Destination Pair: Basic Topologies

In the topology figures presented in this section, a node can transmit data to (and also interfere with) another node (i.e., there exists a link between them), if and only if these nodes are connected by an arrow. Additional (symmetric) interference

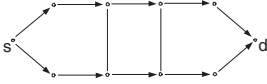


Fig. 1. Bridges between paths

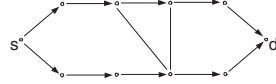


Fig. 2. Triangular structure

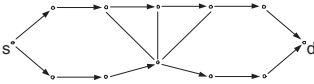


Fig. 3. Claw structure

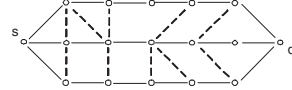
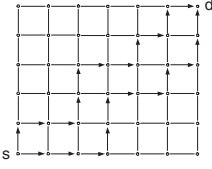
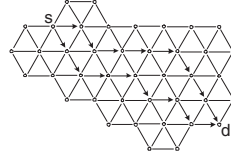


Fig. 4. Example of scenario for 3 node-independent paths of 5 nodes each

relations are indicated by solid lines. The interference and transmission neighborhoods follow accordingly. The paths are assumed to be given and can be seen as provided by a multi-path routing protocol which aims to select node-independent paths. Our interest here is to evaluate the impact of certain interference structures between paths and to find out in which situations using multiple paths is attractive. All the numerical values for the network capacity are computed by solving the program (5) via an exact approach. We note that the greedy approximate algorithm also provides the optimal solution in the cases presented.

Consider a single-path of at least three links from a source s to a destination d . The only interference is the self-interference within the path. Clearly, for capacity optimization all nodes on the path must be utilized in an identical fashion. The network capacity then equals $1/3$ (i.e., the source transmits data $1/3$ of the time), since nodes are not allowed to transmit simultaneously with their neighbors. If the network is shortened to a chain with only one or two links, the capacities would become 1 and $1/2$, respectively.

Next, let us define *independent paths* as paths from source s to destination d that do not interfere with each other except that they share the resources at s and d . The capacity of the SD-flow equals the sum of the flows over all individual paths from s to d . Using only one path, the network capacity clearly equals $1/3$. However, when using both paths, it follows that a capacity of $1/2$ can be attained. Hence, in a situation with interference only at the endpoints of the paths, it is favorable to split the traffic at the source and distribute it over multiple paths. When paths mildly interfere at some points, as in the situation with so-called “bridges” between the paths shown in Fig. 1, a capacity of $1/2$ may still be attained. However, increasing the interference between the paths further may lead to a drastic reduction of the capacity. If a single node is interfering with two nodes on the other path leading to a triangular structure between the paths (see Fig. 2), the capacity becomes $3/7$. Ultimately, when at least one node interferes with three (or even more) nodes (see Fig. 3) on the adjacent path, a claw structure arises and the capacity decreases to $1/3$, the single-path capacity. Thus, in a situation of two paths with at least one

**Fig. 5.** Grid structure**Fig. 6.** Honeycomb structure

heavily interfering node, employing an extra path does not affect the capacity in any meaningful way.

As an example, we consider networks with a more general structure: a grid (or mesh) network (see Fig. 5) and a honeycomb network (see Fig. 6). Each of the intersections corresponds to a node and the edges indicate that node pairs can communicate. Typically, in such networks there exist many equal-length paths between source s and destination d . Employing two independent shortest paths would yield the optimal capacity of $1/2$. One could argue that paths that more closely mimic the straight line between s and d should be preferred as those for example can be found more easily using geographical information, or alternatively, such paths restrict interference to a small corridor between s and d , thus causing less interference with other SD-pairs. We observe that several basic structures, namely triangles, bridges and even claws (see Fig. 6), will then appear between such paths and thus determine the network capacity.

4.2 Single Source-Destination Pair: General Topologies

To assess the quality of our greedy algorithm, we have studied several general topologies. More precisely, we have constructed topologies comprising equal-length, parallel, node-independent paths and randomly generated links between nodes on adjacent paths according to independent Bernoulli experiments with success probability p . A sample scenario for such a topology is provided in Fig. 4 where the solid lines indicate the links on the node-independent paths and the dashed lines the randomly generated links between these paths. The interference and transmission range are set to be equal, i.e., any link in the network induces interference, but can also be used for transmission. For all $j \in \mathcal{N}$, the neighborhoods $\mathcal{N}_{\mathcal{T}}(j)$ and $\mathcal{N}_{\mathcal{I}}(j)$ follow directly from this link set and we have $\mathcal{N}_{\mathcal{T}}(j) = \mathcal{N}_{\mathcal{I}}(j)$. For each scenario, we study the performance of the greedy algorithm by carrying out 50 runs of different random link configurations between the paths.

For moderate-size topologies (i.e., fewer than 20 nodes), we are still able to attain the optimal capacity by means of standard solvers. In Table 1, the average capacity values are presented for several scenarios and compared with the outcomes achieved by the greedy algorithm. Also included are the percentages of runs for which the greedy algorithm deviates more than 5% or 10% from the optimum. In the final scenario (5 paths of 3 nodes, $p = 0.80$), the greedy algorithm deviates more from the optimum, because it only succeeds in finding two independent paths while three such paths are present. However, the results show that, on average, the

Table 1. Capacity results for moderate-size topologies

Scenario		Optimal	Greedy	Err.>5%	Err.>10%	#LP solved
3 paths, 5 nodes	p=0.40	0.518	0.516	0%	0%	4.50
"	p=0.80	0.500	0.500	0%	0%	7.70
4 paths, 4 nodes	p=0.40	0.568	0.566	0%	0%	4.66
"	p=0.80	0.512	0.512	2%	0%	6.38
5 paths, 3 nodes	p=0.40	0.630	0.630	0%	0%	4.80
"	p=0.80	0.602	0.588	18%	10%	6.84

Table 2. Capacity results for large topologies

Scenario		p=0.4		p=0.8	
	Total nr. of nodes	All used	Greedy	All used	Greedy
4 paths, 6 nodes	26	0.564	0.568	0.502	0.512
5 paths, 6 nodes	32	0.616	0.624	0.522	0.590
6 paths, 6 nodes	38	0.654	0.660	0.568	0.612
7 paths, 6 nodes	44	0.684	0.692	0.612	0.646

greedy algorithm performs close to optimal, that its solution rarely deviates far from the optimal capacity, and that it requires limited computational effort (see the last column which indicates the average number of LPs solved per run).

For larger topologies (i.e., greater or equal than 20 nodes), exact solution approaches are no longer computationally feasible. On the contrary, we can rely on the greedy algorithm to find a fast approximation for the capacity. In Table 2, we compare the greedy approximation with the solution for the case that all interference constraints (8) are taken into account and only a single LP problem is to be solved; this corresponds to replacing Eq. (8) by Eq. (14). The comparison shows that for $p = 0.8$ our algorithm yields a much better approximation for the capacity, while for $p = 0.4$ the approximations are similar. We conclude that application of the greedy algorithm is especially valuable in situations in which there is much interference. Another observation from our experiments is that for high values of p , the source forwards traffic via fewer nodes than for low values of p . This suggests that in the case of heavy interference only a few paths need to be used.

4.3 Advanced Scenarios

Many networks are employed in situations where typically more than a single subject wants to transmit data. To this end, we study the network capacity in the situation of two (intersecting) SD-pairs. Regarding the objective function $h(\boldsymbol{\rho}_t, \mathbf{p})$, we do not merely optimize the total capacity (as one pair may consume all the capacity, while the other pair may starve), but we impose additional restrictions on the individual flows. More specifically, we let $h(\boldsymbol{\rho}_t, \mathbf{p}) = \rho_{t,s_1}^{(1)} + \rho_{t,s_2}^{(2)}$, and impose $\rho_{t,s_2}^{(2)} = \frac{1-c}{c} \cdot \rho_{t,s_1}^{(1)}$, $0 \leq c \leq 1$. That means that, e.g., for $c = 1/2$, both flows are equally important, while for $c > 1/2$, SD-pair 1 is prioritized.

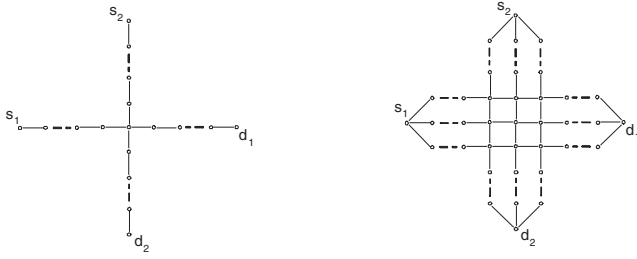


Fig. 7. a. Two single-paths (left), and b. Two multi-paths (right)

We construct two scenarios (along the approach in Sect. 4.2) with paths chosen to be long enough to avoid boundary effects. The capacity is determined using the approximate greedy algorithm. In the first scenario (see Fig. 7a), we consider two intersecting single-path flows. Hence, the central node can be active for at most $1/3$ of the time (as it must share resources with its neighbors), so that the capacity equals $1/3$ independent of the value for c . The second scenario (see Fig. 7b) comprises two intersecting multi-paths which each consist of three node-independent paths. Notice that the value of c influences the capacity in a symmetrical fashion. We discuss here the extreme cases $p = 0$ (independent paths) and $p = 1$ (heavily interfering paths). For $p = 1$, the impact of c is limited as for $c = 1/2$ the capacity equals 0.48, while for $c \rightarrow 0$ it approaches 0.50. Contrary, for $p = 0$, the capacity for $c = 1/2$ is 0.74, while for $c \rightarrow 0$ it is only 0.60. Thus, prioritizing one flow at the expense of the other can eventually be undesirable from a capacity viewpoint in the latter case. Again, we observe that the “multi-path” flow tends to use more paths for $p = 0$ than for $p = 1$.

As a final illustration, in accordance with more realistic applications, we analyze different interference and transmission neighborhoods. We construct scenarios which conform to the approach in Sect. 4.2 with link probabilities $p = 0$ and $p = 1$ (for a single SD-pair). For these choices of p , the neighborhoods $\mathcal{N}_T(j)$ and $\mathcal{N}_I(j)$ are fixed and identical for a given $j \in \mathcal{N}$. Let us denote these neighborhoods by $\mathcal{N}_0(j)$ and $\mathcal{N}_1(j)$, respectively. Next, we define extended neighborhoods as follows. For $p = 0$, let $\mathcal{N}_0^*(j)$ extend $\mathcal{N}_0(j)$ with the three parallel nodes on the two nearest paths and with the two adjacent nodes on the same path. For $p = 1$, let $\mathcal{N}_1^*(j)$ extend $\mathcal{N}_0^*(j)$ with the three parallel nodes of the paths located next to the adjacent paths (whenever present). Thus, we have that $\mathcal{N}_0^*(j) \supseteq \mathcal{N}_0(j)$ and $\mathcal{N}_1^*(j) \supseteq \mathcal{N}_1(j)$. We stress that the exact structure of the neighborhoods is not especially important here. Our intention is to gain some understanding of what might happen in realistic cases. In practice, the exact neighborhoods could be constructed based on network measurements.

We have evaluated scenarios with 1, 3 and 5 node-independent paths consisting of six “single-links” from source to destination. In Table 3, the capacity results (obtained via our greedy algorithm) for the different neighborhoods and link probabilities are presented. There are two main conclusions that can be drawn from our experiments. First, increasing the interference range for a fixed transmission

Table 3. Capacity results for different neighborhoods

Scenario	Neighborhoods	1 path	3 paths		5 paths	
		p=0/1	p=0	p=1	p=0	p=1
Basic scenario	$\mathcal{N}_T(j) = \mathcal{N}_I(j) = \mathcal{N}_{0/1}(j)$	0.333	0.600	0.500	0.714	0.583
Ext. interference	$\mathcal{N}_T(j) = \mathcal{N}_{0/1}(j),$ $\mathcal{N}_I(j) = \mathcal{N}_{0/1}^*(j)$	0.200	0.333	0.250	0.429	0.382
Ext. power	$\mathcal{N}_T(j) = \mathcal{N}_I(j) = \mathcal{N}_{0/1}^*(j)$	0.333	0.500	0.400	0.600	0.529

range has a clear negative effect on the capacity, and this is shown to hold for all scenarios. Second, also increasing both ranges (in an equal fashion) has a (minor) negative effect on the capacity, except for the case of a single path. Further, we observe that the solutions found comprise fewer paths when the interference range is extended; this indicates again that the impact of interference on path selection cannot be ignored.

5 Concluding Remarks

We have first presented a general stochastic framework for the analysis of network performance under interference. We have then developed a generic mathematical model to assess the network capacity in a multi-path environment. A nonlinear programming problem formulation incorporating interference has been introduced, and it has been shown that the optimal network capacity could be obtained by solving an exponential number of linear programming problems. We have presented a greedy algorithm that can closely approximate this optimum by solving only a small number of these linear programmes. Our examples further show that paths do not need to be independent to attain the optimal capacity under interference. Moreover, using multiple paths that moderately interfere appears to be better than using a single path, whereas in situations with heavily interfering nodes, use of multiple paths does not improve the network performance. Finally, more general scenarios have shown the significant impact of interference on the optimal route selection.

An important element of our modelling approach is the construction of interference and transmission neighborhoods. Our model facilitates consideration of any neighborhood of interest; in particular, the neighborhoods do not need to depend on the distance between nodes. This means that by performing certain network measurements before the actual network operation starts neighborhoods can accurately be defined and the capacity of the constructed network can be determined using the techniques presented in this paper.

We have assumed that links in the network are error-free. This is a valid assumption if lossy links are excluded in advance (cf. the approach in e.g. [16]). Otherwise, non error-free links may be incorporated by reducing the transmission rate so as to account for retransmission overhead. The validation of our results for more general networks in which nodes are not identical is among our aims for further research. It is further relevant to study the impact of interference in the context of delay

optimization, because blindly employing multiple independent paths may yield undesirable delays due to the divergence in path lengths.

References

1. Clausen, T. (eds.): P.J.: Optimized Link State Routing protocol (OLSR) (2003)
2. Johnson, D., Maltz, D.: Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, vol. 353 (1996)
3. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: *Proc. of the 2nd IEEE Workshop on Mobile Computing System and Applications*, pp. 90–100 New Orleans, LA, USA (1999)
4. Mueller, S., Tsang, R., Ghosal, D.: Multipath routing in mobile ad hoc networks: Issues and challenges. *Lecture Notes in Computer Science* (2004)
5. Lee, S.J., Gerla, M.: AODV-BR: Backup routing in ad hoc networks. In: *Proc. of WCNC*, Chicago, Illinois, USA (2000)
6. Nasipuri, A., Castañeda, R., Das, S.: Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *ACM/Baltzer Mobile Networks and Applications (MONET) Journal* 6, 339–349 (2001)
7. Lee, S.J., Gerla, M.: Split multipath routing with maximally disjoint paths in ad hoc networks. In: *Proc. of IEEE ICC*, pp. 3201–3205. Helsinki, Finland (2001)
8. Pham, P., Perreau, S.: Increasing the network performance using multi-path routing mechanism with load balance. *Ad. Hoc. Networks* 2, 433–459 (2004)
9. Tsirigos, A., Haas, Z.: Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine* 39(11), 132–138 (2001)
10. Wu, K., Harms, J.: Multipath routing for mobile ad hoc networks. *Journal of Communication and Networks*, vol. 4 (2002)
11. Pearlman, M.R., Haas, Z.J., Sholander, P., Tabrizi, S.S.: On the impact of alternate path routing for load balancing in mobile ad hoc networks. In: *Proc. of MobiHoc*, Boston, Massachusetts, USA (2000)
12. Jain, K., Padhye, J., Padmanabhan, V., Qiu, L.: Impact of interference on multi-hop wireless network performance. In: *Proc. of Mobicom*, San Diego, California, USA (2003)
13. Gupta, R., Musacchio, J., Walrand, J.: Sufficient rate constraints for QoS flows in ad-hoc networks (UCB/ERL Technical Memorandum M04/42, 2004)
14. Litjens, R., van den Berg, H., Boucherie, R.J., Roijers, F., Fleuren, M.: Performance analysis of wireless LANs: an integrated packet/flow level approach. In: *Proceedings of ITC-18*, Berlin, Germany (2003)
15. Schrijver, A.: *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Heidelberg (2006)
16. Padhye, J., Agarwal, S., Padmanabhan, V., Qiu, L., Rao, A., Zill, B.: Estimation of link interference in static multi-hop wireless networks. In: *Proc. of IMC*, Berkeley, CA, United States (2005)