On End-to-end Delay Minimization in Wireless Network under Physical Interference Model

Abstract—**The problem of scheduling transmission in single hop and multi-hop wireless networks with arbitrary topology and physical interference model has been widely studied. However, the focus has been mainly on optimizing the efficiency in transmission parallelization through a minimum frame-length scheduling that, for a given set of traffic demands, provides the smallest number of time-slots for performing different sets of simultaneous and compatible (according to the interference model) transmissions. Unfortunately, this optimal resource reuse efficiency does not in general correspond to the best performance in terms of end-to-end packet delivery delay since multiple frames may be required to complete transfer from source to destination. In this paper we study the problem of minimizing the end-to-end delay in wireless network under Signal to Interference plus Noise Ratio (SINR) constraints, and propose two schemes. The first scheme extends the minimum frame-length approach and minimizes delay over multiple frames. The second scheme directly optimizes delay without the constraint of periodic framing. We propose novel mixed integer programming models for the two schemes and study their properties and complexity. Moreover, we present an efficient heuristic method that provides good quality solutions in short time.**

Index Terms—**Link scheduling, Routing, Multi-hop wireless, SINR model, Optimization, Mathematical Programming.**

I. Introduction

Most of the performance indicators of wireless networks depend on how efficiently radio resources are managed so as to parallelize transmissions on the same communication channel while keeping mutual interference under control. The physical interference model assumes that transmissions can be correctly received if Signal to Interference plus Noise Ratio (SINR) values at receivers are above a threshold. Transmissions that do not meet this constraint must be orthogonalized typically over time using a proper scheduling strategy. In multi-hop wireless networks, communications may involve multiple transmissions on the links along a path from source to destination, and, for delivering a packet, transmissions must obviously occur in sequence.

The problem of scheduling in single hop and multi-hop wireless networks with arbitrary topology has been widely studied [1]–[3]. However, the focus has been mainly on optimizing the efficiency in transmission parallelization through a minimum frame-length scheduling that, for a given set of traffic demands, provides the smallest number of time-slots for performing different sets of compatible (according to the interference model) transmissions [4], [5]. Basically, given a total set of packet transmissions to be performed, these are partitioned into a minimum number of compatible subsets so as to maximize parallelization and resource reuse. Scheduling ensures that each link in the network appears in the different compatible sets a number of times that is sufficient for transmitting all packets. In multi-hop wireless networks, the number of packets to be transmitted per link depends on the routing that can be optimized as well, so as to ensure global network efficiency [6]–[8].

Our work is based on the observation that the main performance indicators that are more directly related to the quality observed by traffic demands, such as rate and delay, cannot be easily mapped to the resource reuse efficiency and that considering these indicators into the optimization process can provide quite better solutions. With minimum frame-length scheduling the sequence of compatible sets in different time slots is not relevant, and it is assumed that frames repeats periodically so as to ensure that packets can be delivered over multi-hop connections using multiple frames if necessary. The rate of a traffic demand is then constant and given by the total number of bits/packets divided by the frame-length. Since the frame length is subject to optimization, the rates tend to be maximized and typically larger than necessary (required by traffic). On the other side, end-to-end delay is not considered and depends on the sequence of time-slots in the frame, several frames may be necessary to deliver a packet from source to destination.

In this paper we study the problem of minimizing the end-to-end delay in wireless networks under the physical interference model. The end-to-end delay is defined as the number of time slots required for delivering a set of packets from their sources to destinations. Minimizing total end-to-end delay and the min-max delay in this paper are equivalent and thus they are used interchangeably.

This problem extends the applicability scenarios of the classical joint routing and scheduling problem to those where packet delivery delay is a critical quality parameter like, for example, in wireless MESH networks supporting real-time traffic, in data gathering networks that concentrate time critical information to a gateway, in alert message broadcasting/multicasting, etc.

We propose and compare two schemes for optimizing the end-to-end delay. *Scheme I* extends the classical minimum frame-length scheduling to include an additional delay minimization and is based on two phases. In the first phase, the minimal frame length and the optimal list of compatible sets are computed. Then, for the second phase, we introduce a mixed integer programming model for ordering the obtained compatible sets in the frame such that the maximal delay over all packets is minimized. The frame is assumed to be periodic and the order of compatible sets constant. We formally prove that the problem in the Phase 2 or ordering compatible sets is

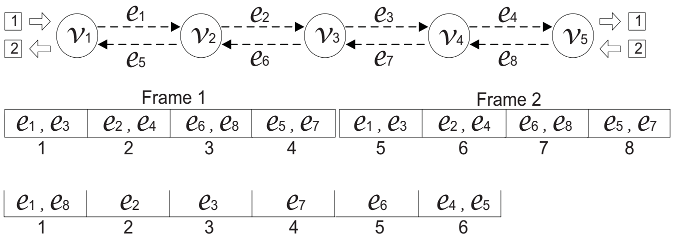


Fig. 1: An illustration for the delay of two schemes

NP-hard. The main advantage of scheme I is that it retains the property of minimum frame-length scheduling of providing constant rates to traffic demands which is particularly important in scenarios with applications requiring fixed bandwidth. On the other side, the limitations imposed by the minimal number of compatible sets and the periodic frame do not allow to fully optimize delay.

In *Scheme II*, the delay is calculated on a slot-by-slot basis without the constraint of a periodic frame. We propose a mixed integer programming formulation of the problem, which overcomes the classical approach of combining routing and scheduling variables for the selection of compatible sets. It selects an ordered sequence of compatible sets considering variables describing which packets have been received (and then can be retransmitted) by each node in previous time slots. The key advantage of Scheme II is that it allows the best performance in terms of end-to-end delay minimization and is particularly suitable for applications requiring timely delivery of a set of given messages. Moreover, it can be easily adapted to changing scenarios where additional requests must be accommodated when the scheduling is already in progress. However, it does not guarantee in general a constant rate unless some kind of periodic optimization is considered. We also prove that the problem of Scheme II is NP-hard. Since it is quite time-consuming to solve this model, we also propose an efficient randomized greedy method.

To illustrate the difference in computing the delay in the two schemes, we provide an example in Fig. 1 where five nodes are evenly distributed along a linear topology and the interference range is two hops. Two packets need to be transmitted, one from node v_1 to node v_5 and the other from node v_5 to node v_1. For Scheme I, the minimal frame length is 4 time slots, as indicated in the figure, and two frames (8 time slots) are needed for the two packets to arrive at their destinations. The packet from node v_1 to node v_5 is delivered in 6 slots, while the other packet in 8 slots. For Scheme II, 6 different compatible sets can be used, as also indicated in the lower part of the figure, and both packets can be delivered in 6 slots.

Finally, we make extensive numerical studies to illustrate the optimal and heuristic solutions and compare the two schemes by varying the number of packets, the SINR threshold and the hop distance between source and destination. Further we show their differences in transmission parallelization and reports the computational efficiency for all proposed models and methods.

The paper is organized as follows. In Section II we review and discuss related work. In Section III we present the system model adopted for problem analysis, while in Section IV we present Scheme I with its two phases for frame length minimization and for delay minimization, and study its properties and complexity. In Section V we present and analyze Scheme II that directly optimizes delay. In Section VI we present our numerical analysis of the proposed schemes, while in Section VII we conclude the paper with final remarks.

II. Related work

One-shot link scheduling in arbitrary wireless networks under the physical interference model, i.e., maximizing the cardinality of a compatible set, has been well studied in the past few years. The paper [1] shows that this problem is NP-hard even when there is no background noise, and then the authors provide some game theoretic results for combing this problem with choosing continuous transmission powers for all nodes. The paper [2] proposes a scheduling algorithm with approximation guarantee independent of the network topology by assuming that the transmission power is uniform for all nodes. The paper [3] proposes distributed algorithms with provable guarantees for both the physical interference model, and the protocol interference model without assuming that transmitters can coordinate with their neighbors in the interference graph.

There has also been quite some work for finding minimum frame-length scheduling. The paper [4] presents a column generation method to minimize the length of the TDMA data frame for the given demands in ad hoc networks. An efficient formulation is given where a column is defined as a compatible set. The paper [5] gives some algorithmic analysis on this problem and provide approximation algorithms with and without power control.

The joint optimization of routing and scheduling for throughput related objectives in wireless networks has been analyzed under different assumptions and models (see [8] for a comprehensive survey). From the modeling and algorithm perspective, the use of the notation of compatible set has been a mainstream method for optimization problems under SINR model. The paper [6] conducts a comprehensive study for joint routing, link scheduling, power control, rate adaptation and channel assignment in wireless mesh networks. A set of mixed integer programming models are provided and the compatible set is also adopted in formulations. The study on flow fairness of wireless mesh networks in [7] also develop optimization models using compatible sets. Several other works utilizing compatible sets can also be found in [9]–[11].

As far as delay minimization is concerned, several different aspects have been considered in recent years. Some papers, like [12], [13], consider the problem under the perspective of queuing delay, which is not the topic of this work. The paper [14] aims to find an optimal link scheduling scheme for minimizing the maximal delay when link bandwidths, a TDMA frame length and routing paths are given. However, the minimal frame length always depends on routing and scheduling (see [15]) and thus this work considers a more restrictive scenario with respect to ours. Moreover, the mixed integer programming formulated in [14] is based on the conflict graph (protocol interference model) which is not easy to be extended to the SINR model. The model in [16] considers SINR constraints and routing, but the model considers mono-directional traffic to a gateway and cannot be applied to an arbitrary traffic demand like ours. Following the idea of [14], other mixed integer programming models based on conflict graph are developed in [17], [18]. The paper [19] studies the problem of minimizing the end-to-end delay for general multi-hop wireless networks. Two models are provided when routing is fixed in advance and is combined with link scheduling respectively. Again, their models use conflict graph.

III. System Model

We model a wireless network by means of a bidirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{A}$ is the set of links. The link set $\mathcal{A}$ has element (i, j), if and only if the signal-to-noise ratio (SNR) condition is satisfied, i.e., $\frac{P_i g_{ij}}{\eta} \geq \gamma$; here, P_i is the transmitting power of node i, η is the noise power, g_{ij} is the gain between nodes i and j, and γ is the threshold. The set of neighbor nodes of i is denoted by $\Gamma(i)$. These are the nodes that directly can reach i or be reached by i, i.e., $\Gamma(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{A} \text{ or } (j, i) \in \mathcal{A}\}$.

A set of links can be active simultaneously in a time slot, if and only if the SINR of each of the links meets the threshold (a single modulation and coding scheme, and then a single SINR threshold, is considered). For compatible set c, we use Δ_{ij}^c as an indicator parameter to represent whether or not link (i, j) is in transmission in c, and we use indicator parameter δ_{ic} to represent whether or not node i is transmitting in compatible set c. The corresponding SINR condition is formulated below.

$$\frac{P_i g_{ij}}{\eta + \sum_{k \neq i: \delta_{kc}=1} P_k g_{kj}} \geq \gamma, \forall (i, j) \in \mathcal{A} : \Delta_{ij}^c = 1 \qquad (1)$$

We denote by $\mathcal{S}$ a given set of packets. Each packet $s \in \mathcal{S}$ is associated with an origin node o_s and a terminate node d_s. The optimization problem consists in minimizing the end-to-end delay of sending all packets, possibly via multiple hops, to their respective destinations. The notations used in this paper is summarized in Table I.

TABLE I: A summary of notations

Notation	Description
$\mathcal{V}$	the set of nodes
$\mathcal{A}$	the set of links
$\mathcal{S}$	the set of packets
o_s	the origin of packet s
d_s	the destination of packet s
$\mathcal{L}$	the set of all compatible sets
T	any valid upper bound for the delay
$\overline{T}$	the minimal frame length
T_I	the optimal delay of Scheme I
T_{II}	the optimal delay of Scheme II
T_h	the delay computed by the randomized greedy method
P_i	the transmitting power of node i
g_{ij}	the power gain between nodes i and j
η	the noise power
γ	the SINR threshold
$\Gamma(i)$	the set of neighboring nodes of node i
Δ_{ij}^c	indicator parameter that equals one if link (i, j) is in compatible set c, and zero otherwise
δ_{ic}	indicator parameter that is one if node i is transmitting in compatible set c, and zero otherwise

IV. Solution Driven by Minimum Frame-Length Scheduling

In the conventional scheduling schemes, a.k.a. as minimum frame-length scheduling, the objective is to minimize the total number of time slots required for transmissions. As was discussed in Section I, this may be sub-optimal from the delay standpoint. However, the compatible sets resulted from minimizing the schedule length serve as an intuitive starting point, for which end-to-end delay is then accounted. Recall that the order of compatible sets is of no significance in minimum frame-length scheduling. Thus the compatible sets derived from minimum frame-length scheduling need to be permutated in a way that minimizes the delay metric. Moreover, there is an inherent assumption of periodicity in minimum frame-length scheduling. Thus, once the order of compatible sets is set, the ordered frame repeats itself as long as not all packets have arrived the destinations.

From the above discussion, the approach based on minimum frame-length scheduling consists in two phases. The first phase solves minimum frame-length scheduling for the given packets and thereby produces a set of compatible sets, such that the number of compatible sets is minimized. Then, in the second phase, the task is to optimize the order of the compatible sets to form an ordered frame, along with the routing of each individual packet, such that the overall delay is minimized with frame periodicity. We remark that routing decision does appear in minimum frame-length scheduling as well. However, we do not include the individual route of each packet in the optimization in the second phase as the routing in phase one may be sub-optimal for the dealy metric.

A. Phase 1 – Finding minimal frame length

The problem of minimum frame-length scheduling has been well studied (see Section II). Let $\mathcal{L}$ be set of all compatible sets. The optimization formulation used by many previous works is shown in (2). The variables are defined as follows.

- t_c: integer variable, representing the number of time slots using compatible set $c, c \in \mathcal{L}$.
- f_{ij}^s: binary variable that equals one if packet s passes through link (i, j), and zero otherwise.

$$\text{minimize } \overline{T} = \textstyle\sum_{c \in \mathcal{L}} t_c \qquad (2a)$$

$$\text{subject to: } \sum_{j \in \Gamma(i)} (f_{ij}^s - f_{ji}^s) = \begin{cases} 1, \ i = o_s \\ -1, \ i = d_s \\ 0, \text{ otherwise} \end{cases}, \ s \in \mathcal{S} \qquad (2b)$$

$$\textstyle\sum_{c \in \mathcal{L}} \Delta_{ij}^c t_c \geq \sum_{s \in \mathcal{S}} f_{ij}^s, \ (i, j) \in \mathcal{A} \qquad (2c)$$

$$t \in \mathbb{Z}^+, f \in \mathbb{B} \qquad (2d)$$

The objective (2a) is to minimize the total number of time slots, denoted by $\overline{T}$. Constraints (2b) impose flow conservation that represents route seletion. Constraints (2c) assure that the sufficiently many compatible sets including link (i, j) are selected to accommodate the packets to be routed via the link.

An effective solution approach for (2) is to solve its linear relaxation by column generation, which has been shown to produce solutions being very close to optimum [4]. Column generation iteratively constructs compatible sets. Let $\mathcal{L}^* \subset \mathcal{L}$ be the current set of compatible sets present in (2). The linear relaxation of (2) for $\mathcal{L}^*$ is also known as the restricted master problem. Denote by π^*_{ij} the optimal dual variables corresponding to constraints (2c) of the restricted master problem. The procedure of column generation consists in finding a compatible set $c \in \mathcal{L} \setminus \mathcal{L}^*$, such that $\sum_{(i,j)\in\mathcal{A}} \pi^*_{ij}\Delta^c_{ij} > 1$. Then adding such a compatible set to the restricted master problem will potentially improve (2a). For further details, see e.g. [20], [21]. The problem for generating a compatible set is formulated below. This is also known as the pricing problem in column generation.

- X_i: binary variable that equals one if node $i, i \in \mathcal{V}$, is transmitting, and zero otherwise;
- Y_{ij}: binary variable that equals one if link $(i,j), (i,j) \in \mathcal{A}$, is active, and zero otherwise.

$$\text{maximize } \sum_{(i,j)\in\mathcal{A}} \pi^*_{ij} Y_{ij} \quad (3a)$$
$$\text{subject to: } \sum_{j\in\Gamma(i)} (Y_{ij} + Y_{ji}) \le 1,\ i \in \mathcal{V} \quad (3b)$$
$$\sum_{j\in\Gamma(i)} Y_{ij} = X_i,\ i \in \mathcal{V} \quad (3c)$$
$$P_i g_{ij} Y_{ij} + M_{ij}(1 - Y_{ij}) \ge \gamma(N + \sum_{k\neq i} P_k g_{kj} X_k),\ (i,j) \in A \quad (3d)$$
$$X, Y \in \mathbb{B} \quad (3e)$$

The objective (3a) is to generate a maximum weighted compatible set. Constraints (3b) make sure that a node can either transmit to or receive from at most one neighbor. Constraints (3c) state that a node i is transmitting if and only if one of its outgoing links is active. Constraints (3d) represent the SINR requirements, where $M_{ij} = \gamma(N + \sum_{k\neq i} P_{kj})$, with the effect when link (i,j) is inactive (i.e., $Y_{ij} = 0$), the constraint becomes void.

We reuse the notation $\mathcal{L}^*$ for the final, optimized set of compatible sets by column generation. The corresponding t-variable may be factional, however. To arrive at an integer solution, the common approach is to solve (2) with $\mathcal{L}^*$ as the given set of compatible sets. We denote by $t^*_c, c \in \mathcal{L}^*$ the resulting integer solution.

For single-hop networks, i.e., the route for each packet involves a direct transmission from the source to the destination, the master problem of finding minimal frame length can be written as a linear programming (4) as follows, without he loss of integrality, as stated by Theorem 1.

$$\text{minimize } \overline{T} = \sum_{c\in\mathcal{L}} t_c \quad (4a)$$
$$\text{subject to: } \sum_{c\in\mathcal{L}} \Delta^c_{ij} t_c \ge 1, \quad i = o_s, j = d_s, s \in \mathcal{S} \quad (4b)$$
$$t \ge 0 \quad (4c)$$

Theorem 1. *The optimal solution of* (4) *is integral.*

Proof. The incidence matrix of (4) is $B = [b_{sc}]_{|\mathcal{S}|\times|\mathcal{L}|}$, with $b_{sc} = \Delta^c_{o_s d_s}$ that equals to zero or one. Thus, the determinant of every square submatrix of B is either zero or one, implying that matrix B is totally unimodular. Moreover, the right-hand side of (4b) are all ones. From these observations, the polytope defined by (4) have integer soltuions as the extreme points [22], and the theorem follows. □

The above result does not mean that minimum frame-length scheduling for single-hop networks can be solved in polynomial time. The reason is that the pricing problem remains NP-hard [23].

B. Phase 2 – Finding optimal permutation

Given the compatible sets $\mathcal{L}^*$ and their corresponding time slots $t^*_c, c \in \mathcal{L}^*$, the task of Phase 2 is to find the optimal permutation of the elements of $\mathcal{L}^*$, along with optimizing the route of each packet. We present the following mixed integer programming formulation for the problem.

The frame length is $\overline{T} = \sum_{c\in\mathcal{L}^*} t^*_c$ time slots. We need to assign each time slot a compatible set. Note that when the order is fixed for a frame, this frame should be repeated for computing the delay. Let K denote any valid upper bound of how many times the frame will be repeated at optimum, and let $T = K\overline{T}$ (K is an integer), and $\mathcal{T} = \{t : 1 \le t \le T\}$. The integer optimization formulation is given in (5), with the following variable definitions.

- λ_t: binary variable that is one if time slot t is used and zero otherwise, $t \in \mathcal{T}$
- x^t_{is}: binary variable that is one if node i is transmitting packet s in time slot t and zero otherwise, $i \in \mathcal{V}$, $s \in \mathcal{S}$, and $t \in \mathcal{T}$.
- y^t_{is}: binary variable that is one if packet s is at node i by time slot t and zero otherwise, $i \in \mathcal{V}$, $s \in \mathcal{S}$, and $t \in \mathcal{T} \cup \{0\}$.
- z^t_c: binary variable that is one if compatible set c is assigned to time slot t and zero otherwise, $c \in \mathcal{L}^*, t \in \mathcal{T}$
- w^t_{isc}: auxiliary binary variable that represents the product $\delta_{ic} \times x^t_{is} \times z^t_c$, $i \in \mathcal{V}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$, and $c \in \mathcal{L}^*$. Note that δ_{ic} is constant.

The objective (5a) is to minimize the total number of time slots before all packets arrive their respective destinations. By constraints (5b) time slot t is used if there is transmission in this time slot, and a node can transmit at most one packet in a time slot. Constraints (5c) assure consecutive use of time slots, that is a time slot can be used only if the previous one has been used. Constraints (5d) state the initial and final condition of the packet locations, namely all packets are at their source node at the beginning and at their destinations by the end, respectively. By (5e), node i can transmit packet s at time slot t, only if this packet is available to node i in time slot $t-1$, and by (5f), if packet s is available to node i in time slot $t-1$, then this remains the case in time slot t as well as later time slots. The next three sets of constraints, (5g) – (5i), together express the relation between varibles w and variables x, z, i.e., $w^t_{idc} = \delta_{ic} \times x^t_{id} \times z^t_c$, though in a linear fashion. By (5j), packet s arrives node j in time slot t but not earlier, i.e., $y^t_{js} - y^{t-1}_{js} = 1$, if there is at least one node transmitting packet s to j and the compatible set containing such a link (i,j) is assigned to time slot t. By (5k), node i can transmit in time slot t only if it is in the compatible set assgined to time slot t.

Constraints (5l) ensure that, if compatible set c is assigned to time slot t, then this must be the case for all repeated frames. In other words, the order of compatible sets does not change from one frame to the next. Constraints (5m) state the fact that the total number of time slots used by a compatible set (in any frame) equals to t_c^* given by Phase 1. Finally, (5n) ensures that a time slot can accommodate no more than one compatible set.

$$\text{minimize } T_I = \textstyle\sum_{t\in\mathcal{T}} \lambda_t \tag{5a}$$
$$\text{subject to: } \textstyle\sum_{s\in\mathcal{S}} x_{is}^t \le \lambda_t,\ i\in\mathcal{V}, t\in\mathcal{T} \tag{5b}$$
$$\lambda_{t+1} \le \lambda_t,\ t\in[1,T-1] \tag{5c}$$
$$y_{o_s s}^0 = 1,\quad y_{d_s s}^T = 1,\quad y_{vs}^0 = 0,\ v\in\mathcal{V}\setminus\{o_s\}, s\in\mathcal{S} \tag{5d}$$
$$x_{is}^t \le y_{is}^{t-1},\ i\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S} \tag{5e}$$
$$y_{is}^{t-1} \le y_{is}^t,\ i\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S} \tag{5f}$$
$$w_{isc}^t \le \delta_{ic} x_{is}^t,\ i\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S}, c\in\mathcal{L}^* \tag{5g}$$
$$w_{isc}^t \le \delta_{ic} z_c^t,\ i\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S}, c\in\mathcal{L}^* \tag{5h}$$
$$w_{isc}^t \ge \delta_{ic}(x_{is}^t + z_c^t - 1),\ i\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S}, c\in\mathcal{L}^* \tag{5i}$$
$$\textstyle\sum_{i\in\Gamma(j)} \delta_{jc} w_{isc}^t \ge y_{js}^t - y_{js}^{t-1},\ j\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S}, c\in\mathcal{L}^* \tag{5j}$$
$$\textstyle\sum_{c\in\mathcal{L}^*} \delta_{ic} z_c^t \ge \sum_{s\in\mathcal{S}} x_{is}^t,\ i\in\mathcal{V}, t\in\mathcal{T} \tag{5k}$$
$$z_c^t = z_c^{t+k\overline{T}},\quad t\in[1,\overline{T}], k\in[1,K-1], c\in\mathcal{L}^* \tag{5l}$$
$$\textstyle\sum_{t\in[1,\overline{T}]} z_c^t = t_c^*,\quad c\in\mathcal{L}^* \tag{5m}$$
$$\textstyle\sum_{c\in\mathcal{L}^*} z_c^t = 1,\quad t\in[1,\overline{T}] \tag{5n}$$
$$x, y, w, z, \lambda \in \mathbb{B} \tag{5o}$$

For complexity, we present the following result for optimal permutation.

Theorem 2. *Finding optimum permutation is NP-hard.*

Proof. We construct problem reduction using the job shop scheduling problem [24]. In the job shop scheduling problem, n jobs of varying sizes and m identical machines are given. An operation refers to processing of a job on a machine, and a job may require multiple operations in order to be accomplished. Denote the set of operations by $\mathcal{O} = \{(i,j) | j = 1,2,...,n; i\in M^j \subset M := \{1,2,...,m\}\}$ where M^j is the set of machines having operations for job j. Let t_{ij} be the completion time of the operation (i,j). The objective is to minimize the makespan which is the total time for finishing all jobs, i.e., min $\max_{(i,j)\in\mathcal{O}} t_{ij}$, subject to the constraints that two different jobs cannot be processed in a machine at the same time, and a job cannot be processed by two different machines at the same time.

Given any job shop scheduling problem, we define a special case of finding optimum permutation, for given packet routes with one single compatible set containing all links used by the routes. Each packet corresponds to a job and each link in the compatible set corresponds to a machine, i.e., $n = |\mathcal{S}|, m = |\mathcal{A}|$. The set of operations define the routes, $\mathcal{O} = \{e | e \in R_s, s\in\mathcal{S}\}$ and the completion time for each operation is $t_e, e\in\mathcal{O}$. Each link cannot send more than one packet at a time, and a packet will not by sent on more than one link at the same time. The objective is minimizing delay becomes hence min $\max_{e\in\mathcal{O}} t_e$. Thus solving the special case of optimal permutation gives the optimum to the job shop scheduling problem, and the theorem follows. □

We remark that the problem of finding optimum permutation is solved in polynomial time when there is only one packet. In this case, The optimal route is the shortest path between the source and the destination, and the optimal permutation follows the order of links of the path. Also, we note that for a single-hop network, the minimum frame-length schedule is optimal and there is no need of optimizing the permutation as it does not affect the delay.

V. Scheme II – Delay-Driven Scheduling

In this scheme we minimize the delay by jointly optimizing the composition and sequence of compatible sets, as well as the routing of each packet. The restriction of periodicity is no longer present and hence the notion of frame is of no significance. We present a mixed integer programming model for this problem and then propose an efficient randomized greedy method.

A. Optimization model

As before, we use T to represent any valid upper bound for delay, with $\mathcal{T} = \{1, 2, \ldots, T\}$. The integer programming formulation is provided below, in which the variables, λ, x, y have the same definitions as in (5).

$$\text{minimize } T_{II} = \textstyle\sum_{t\in\mathcal{T}} \lambda_t \tag{6a}$$

subject to: (5b) – (5f)

$$\textstyle\sum_{i\in\Gamma(j)} x_{is}^t \ge y_{js}^t - y_{js}^{t-1},\quad j\in\mathcal{V}, t\in\mathcal{T}, s\in\mathcal{S} \tag{6b}$$
$$\textstyle\sum_{s\in\mathcal{S}} (y_{is}^t - y_{is}^{t-1}) + \sum_{s\in\mathcal{S}} x_{is}^t \le 1,\quad i\in\mathcal{V}, t\in\mathcal{T} \tag{6c}$$
$$P_i g_{ij} x_{is}^t + M_{ijs}^t (1 + y_{js}^{t-1} - y_{js}^t) + M_{ijs}^t (1 - x_{is}^t) \ge \gamma(\eta + \textstyle\sum_{k\ne i} P_k g_{kj} \sum_{s\in\mathcal{S}} x_{ks}^t),\quad (i,j)\in\mathcal{A}, t\in\mathcal{T}, s\in\mathcal{S} \tag{6d}$$

The objective (6a), as in (5), is to minimize the total number of time slots. Constraints (5b) – (5f) in (5) remain present. Constraints (6b) assure that node j can receive packet s at time slot t, i.e., $y_{js}^t - y_{js}^{t-1} = 1$, only if at least one of its neighbors is transmitting packet s at time slot t. By constraints (6c), node i cannot transmit and receive at the same time.

Constraints (6d) express the SINR requirement, where $M_{ij}^t = \gamma(\eta + \sum_{k\ne i} P_k g_{kj})$. If node j receives packet s from node i in slot t, i.e., $y_{js}^t - y_{js}^{t-1} = 1$ and $x_{is}^t = 1$, then (6d) ensures the SINR threshold is met. Otherwise, either node j does not receive packet s from any node ($y_{js}^t - y_{js}^{t-1} = 0$), or node i does not transmit packet s ($x_{is}^t = 0$), then the constant M_{ij}^t will make inequality hold though void. If node i does not transmit packet s at time slot t, i.e., $x_{is}^t = 0$, the first term in the left-hand side of (6d) equals to 0 while the third term is M_{ij}^t. In this case, $y_{js}^t - y_{js}^{t-1}$ can take either one or zero, depending on whether there are some other neighbor nodes of j that can transmit packet s successfully to j. The third term in the left hand side of (6d) is necessary to make sure that node j is able to receive packet s only if there exists at least one neighbor that can transmit packet s successfully to j. An example is given in Figure 2. In the figure, node i can transmit to node j successfully while node k cannot. We need the third term in (6d) to assure that the inequality remains valid when $x_{ks}^t = 0$ and $y_{js}^t - y_{js}^{t-1} = 1$ for link (k, j).

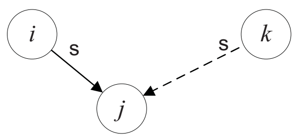


Fig. 2: A transmission scenario.

Theorem 3. *The global optimum in delay is bounded by* $\overline{T}$ *and* T_I*, i.e.,* $T_{II} \in [\overline{T}, T_I]$.

Proof. In Phase 1 of Scheme I, a packet can be transmitted over a link successfully if the half-duplex constraint (3b) and SINR constraint (3d) are satisfied. In Scheme II, in addition to these constraints, formulated by (6c) and (6d), respectively, a packet can be transmitted over a link only if the packet has been buffered in the transmitting node of the link, see (5e). Thus any feasible solution of Scheme II is also feasible for Phase 1 of Scheme I, and $\overline{T}$ is a lower bound of T_{II}. For Phase 2 of Scheme I, apart from the three constraints mentioned above, represented as the conditions in forming compatible sets and (5e)), there is another constraint that a node can send a packet in a time slot only if this node is transmitting in the compatible set assigned to the time slot, see (5k). Thus any feasible solution of Phase 2 of Scheme I is also feasible for Scheme II, and T_I is an upper bound for T_{II}. □

Next, we provide the insight that delay minimization reduces to minimum frame-length scheduling for single-hop network topology. Consequently, minimizing delay is at least as hard as minimizing scheduling length.

Theorem 4. *The delay minimization problem reduces to minimum frame-length scheduling for single-hop topology, and is hence* NP*-hard.*

Proof. For single-hop networks, each packet requires exactly one transmission. Therefore, the overall delay is determined by how many time slots in total are necessary to facilitate one transmission per link. The problem then reduces minimum frame-length scheduling, which is NP-hard [4]. □

We remark that (6) does not restrict the recipient of a packet to be one neighbor of the transmitting node. Multiple neighbors can all receive the packet as long as the SINR constraint is satisfied for the corresponding links. This is a feature of (6), namely, the model is applicable to the scenario of multicast, where some packets have multiple destinations. Such a generalization does not apply for Scheme I. Note that for broadcast, the delay minimization problem remains NP-hard even there is only one single packet [25].

B. A randomized greedy method

For large-scale scenarios, it is time-consuming to solve the model (6) for global optimality. In this section we propose an efficient greedy method for solving delay minimization.

As input to the method, there is a set of candidate compatible sets that admits at least one path from the source to the destination of each packet. The core of the method is the selection of compatible sets to form a complete sequence, ensuring all packets arrive at their respective destinations and targeting minimum delay. The selection of compatible set works in a slot-by-slot basis. For each slot, weights are given to the candidate compatible sets based on the current locations of all packets. The weight of a compatible set estimates how close the packets are from their destination, if the set is selected a time slot. Compatible sets with small weights are given preference in the selection. For robustness, the method adopts randomization. That is, the selection is performed randomly among all compatible sets for which the weights are below a threshold. With embedded randomization, the method consists in multiple runs of sequentially selecting compatible sets, and returning the sequence with the smallest overall delay. We denote the number of runs by N.

In the following as well as for simulation results, we use the $\mathcal{L}^*$, the set of compatible sets obtained in Phase 1 of Scheme I, as the candidate set for the randomized greedy method. Note that $\mathcal{L}^*$ is not the only option for the input list of compatible sets and other heuristic approaches can be considered. We use $\mathcal{C}_c, c \in \mathcal{L}^*$ to denote the links included in compatible set c, i.e., $\mathcal{C}_c = \{(i, j) : \Delta_{ij}^c = 1\}$. Let $\mathcal{V}_s$ denote the set of nodes having packet $s \in \mathcal{S}$ by the time slot in question. We use m_v^s to denote the shortest-path distance, in the number of hops, for packet s from node v to destination d_s. This shortest-path computation is applied to a directed graph including only links that appear in $\mathcal{L}^*$. We introduce w_s to represent the weight of packet s and W_c to represent the weight of compatible set c. The randomized greedy method is described in Algorithm 1.

In Step 3, the value of parameter α depends on the iteration number n. The function $f(n)$ is defined as $f(n) = 0.1$, if $n \le \lceil \frac{N}{3} \rceil$; $f(n) = 0.4$, if $\lceil \frac{N}{3} \rceil < n \le 2\lceil \frac{N}{3} \rceil$; $f(n) = 0.7$, if $2\lceil \frac{N}{3} \rceil < n \le 3\lceil \frac{N}{3} \rceil$.

In Steps 5-11, the algorithm computes the weight for each compatible set. The function $SP((i, j))$ returns a packet randomly selected among those available at node i. There are two conditions in the selection. First, the selected packet is not in node j. Second there exists a path from node j to the destination of this packet. If there are no packets satisfying both conditions, $SP((i, j))$ returns -1. Set $\mathcal{B}$ consists of packets selected from the transmitters of all links in a compatible set.

For each packet in $\mathcal{B}$, there may exist several nodes that can transmit this packet in current time slot if the packet is at this node and the node is a transmitter in the considered compatible set. Then the weight of this packet is computed as the minimum value over all minimum hop-based shortest distances from neighbors (should be receivers in the considered compatible set) of all those nodes (see step 9). For any packets not in $\mathcal{B}$, the weight is the minimum hop-based shortest distance among all nodes having the packet to the destination of the packet (see Step 10).

In Steps 12-13, a compatible set is randomly selected among all compatible sets whose weights are smaller than the

Algorithm 1 Randomized greedy algorithm

Input: $\{\mathcal{C}_c, c \in \mathcal{L}^*\}$, $\{(o_s, d_s), s \in \mathcal{S}\}$, N

1: $n = 0$, $\{\mathcal{V}_s : \mathcal{V}_s = \{o_s\}, s \in \mathcal{S}\}$, $T_h = T$.
2: **while** $n < N$ **do**
3: $t = 0$, $\alpha = f(n)$
4: **while** there are any packets not at its destination **do**
5: **for** $c \in \mathcal{L}^*$ **do**
6: $\mathcal{B} = \emptyset$
7: **for** $(i, j) \in \mathcal{C}_c$ **do**
8: Randomly select a packet H_i from node i, i.e., $H_i = SP((i, j))$; if $H_i \neq -1$, $\mathcal{B} = \mathcal{B} \cup \{H_i\}$
9: $w_s = \min_{(i,j)\in\mathcal{C}_c, i\in\mathcal{V}_s} m^s_j, s \in \mathcal{B}$
10: $w_s = \min_{v\in\mathcal{V}_s} m^s_v, s \in \mathcal{S} \setminus \mathcal{B}$
11: $W_c = \sum_{s\in\mathcal{S}} w_s$
12: Compute threshold $R = \min_{c\in\mathcal{L}^*} W_c + \alpha \times (\max_{c\in\mathcal{L}^*} W_c - \min_{c\in\mathcal{L}^*} W_c)$
13: Randomly select a compatible set c^* from the set $\{c : W_c \leq R, c \in \mathcal{L}^*\}$
14: Define the set of active nodes $\mathcal{M}_{c^*} = \{i : (i, j) \in \mathcal{C}_{c^*}, H_i \neq -1\}$
15: **for** $i \in \mathcal{M}_{c^*}$ **do**
16: For any $k \in \Gamma(i)$, if the SINR constraint is satisfied for link (i, k) with respect to the interfering nodes in $\mathcal{M}_{c^*}$), then $\mathcal{V}_{H_i} = \mathcal{V}_{H_i} \cup \{k\}$
17: $t = t + 1$
18: If $T_h > t$, $T_h = t$
19: **return** T_h

threshold. The threshold is computed in Step 12. If $\alpha = 1$, the compatible set is selected completely randomly while if $\alpha = 0$, the selected compatible set is always the one having the smallest weight. Parameter α changes over the number of iterations in order to reach a balance between randomness and greediness.

In Step 14-16, the locations of all packets are updated. Every active transmitter $i \in \mathcal{M}_{c^*}$ will transmit packet H_i to its neighbors $k \in \Gamma(i)$ for which the SINR constraint is satisfied, i.e., $P_i g_{ik}/(\eta + \sum_{m\in\mathcal{M}_{c^*}\wedge m\neq i} P_m g_{mk}) \geq \gamma$.

VI. Numerical Study

The aim of the numerical analysis presented here is three-fold. Firstly, to facilitate understanding different behaviors of the two schemes, we show the structure of their optimal solutions for a grid network. Then, we compare the two schemes under several factors and show the differences of the two schemes in terms of transmission parallelization. Finally, we illustrate the computational efficiency of the proposed models and the randomized greedy method through extensive experiments.

In all experiments, the noise power is $\eta = 10^{-13}$ W and the transmission power for each node is $P = 0.1$ W. The power gain between node i and node j is $g_{ij} = d^{-4}_{ij}$ where d_{ij} is their distance. Given a SINR threshold γ, the maximal transmission distance can be computed as $L = (P/(\gamma \times \eta))^{\frac{1}{4}}$, which is derived from SNR condition. The parameter K in Phase 2 of Scheme I is set to be 5 and the number of iterations N for the randomized greedy method is set to be 500. All mixed integer programming and linear programming have been solved by the solver Gurobi [26], and the computations have been executed on an Intel Xeon E5620 at 2.4 GHz and with 8 GB RAM under Linux.

A. An illustrative example for optimal solutions of two schemes

To illustrate optimal solutions, we take a grid network in Fig. 3 as an example, in which the length of the edge is 250 m. We consider two packets, which are from node 0 to node 11 and from node 3 to node 8, respectively. The SINR threshold is set to be $\gamma = 5$, for which the maximum supporting distance $L = 376$ m. The optimal routing paths are plotted in Fig. 3 while the optimal link scheduling is shown in Table II in which the frame is repeated for Scheme I while there are no repeated frames in Scheme II. In Scheme I , the minimal frame length is 5 time slots and thus the compatible set in time slot 6 is the same as that in time slot 1.

We can see that the delay of Scheme II is smaller than that of Scheme I. The routing paths of each packet for the two schemes are different since Scheme II finds routing paths with more emphasis on reducing the delay than Scheme I.

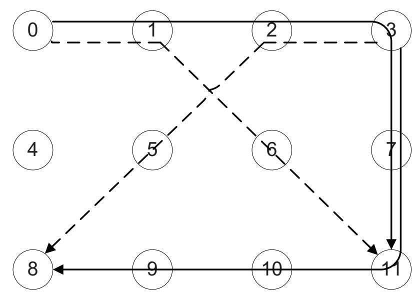


Fig. 3: A grid network and optimal routing paths: solid lines represent Scheme I and dashed lines represent Scheme II

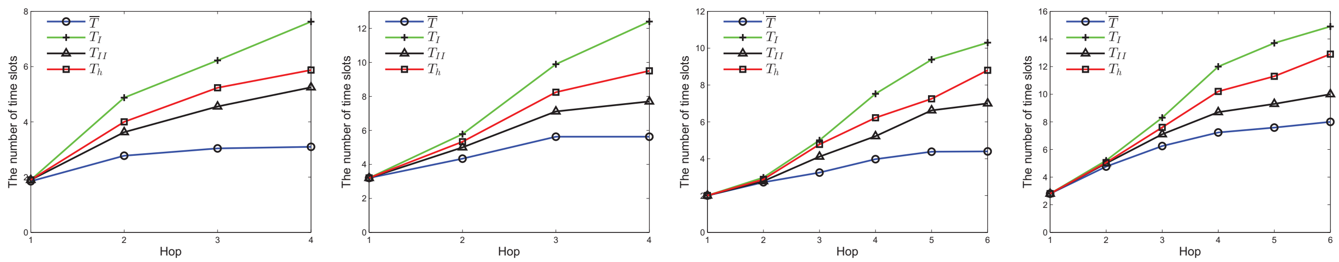
TABLE II: Optimal solutions of the two schemes

Scheme I	1: (3,7), (1,4) → 2: (2,11), (9,8), (7,11) → 3: (0,1), (3,2), (8,9), (11,10) → 4: (1,2), (4,8), (11,10) → 5: (6,4), (2,3), (10,9) → 6: (3,7), (1,4) → 7: (2,11), (9,8), (7,11)
Scheme II	1: (0,1), (3,2) → 2: (2,5) → 3: (5,8) → 4: (1,6) → 5: (6,11)

B. Comparison of the two schemes

We compare here the optimal delay computed by Scheme I and Scheme II, i.e., T_I and T_{II}, as well as the minimal frame length ($\overline{T}$) and the delay computed by the randomized greedy method for Scheme II (T_h). We also show how values depend on the SINR threshold, the number of packets, and the hops per packet (minimum number of hops from its source to its destination).

The comparisons are made by using networks which are generated by randomly distributing nodes in a square area of 1000 m × 1000 m. Delay values for 15-nodes networks are shown in Fig. 4 as a function of the number of hops per packet and for different values of the total number of packets and



(a) Two packets, $\gamma = 5$. (b) Five packets, $\gamma = 5$. (c) Two packets, $\gamma = 10$. (d) Five packets, $\gamma = 10$.

Fig. 4: The comparison of the two schemes under different factors

SINR threshold. For each value of the number of hops per packet, we randomly selected source-destination pairs with this distance and averaged results obtained over 10 instances.

For all sub-figures, we can see that $T_I > T_{II} > \overline{T}$, as indicated by Theorem 3. When the packet is one-hop, $T_I = T_{II} = \overline{T}$, which is consistent with our analysis for single-hop networks in Section IV, i.e., for single-hop network, the minimum frame-length schedule is optimal for computing the delay. The randomized greedy method also provides the same number of time slots in this case since in our implementation, it uses the list of compatible sets in the optimal solution of minimum frame-length schedule.

Considering the gap between T_I and T_{II} in terms of $(T_I - T_{II})/T_I \times 100\%$, it increases with the number of hops. Take Fig. (4a) as an example, the gap corresponding to 2,3,4 hops are around 34%, 37%, and 45% respectively. This gap also increases with the increasing number of packets, for example, the gap for 4 hops in Fig. (4a) is 45% while in Fig. (4b) is 61 %. This indicates that Scheme II will perform much better than Scheme I when the number of packets becomes large. Further, the gap becomes larger when the SINR threshold decreases, for example, the gap for 4 hops is 42% in Fig. (4c) while 45% in Fig. (4a), and 30% in Fig. (4d) while 61% in Fig. (4b).

We can observe that the delay computed by the randomized greedy method is much lower than the delay of Scheme I which is the upper bound of Scheme II, and is very close the optimum of Scheme II. This shows the effectiveness of our proposed heuristic method.

We also compared the two schemes in terms of transmission parallelization, measured according to the metric: average number of transmissions per compatible set. It is calculated as the sum of the number of links over all compatible sets divided by the number of compatible sets used in the optimal solution. Note that links in each time slot of Scheme II make up a compatible set. For example, the average number of transmissions per compatible set for two schemes in Table II are $(2{+}3{+}4{+}3{+}3)/5 = 3$ and $6/5 = 1.2$ respectively. Results obtained for different size networks are shown in Fig. 5. We set that the number of packet to 3 and the SINR threshold to 5. Values are averaged on 10 networks.

We can see that Scheme I has much larger average number of transmissions per compatible set than Scheme II, especially for larger networks. The reason is that Scheme II is driven by the delay and selects compatible sets that include only links for moving packets hop-by-hop to destination. While Scheme I optimizes spatial reuse maximizing the number of links per compatible set and at the cost of a larger delay.

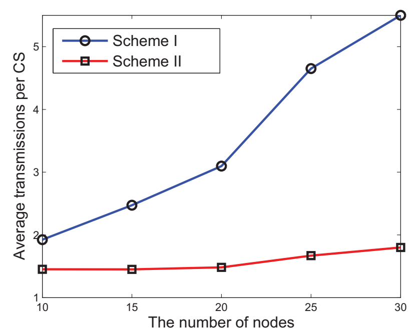


Fig. 5: Average number of transmissions per compatible set.

C. Computational efficiency of all methods

Finally, we present some results illustrating the delay (in number of time slots) and the running time provided by model (2), model (5), model (6) and the randomized greedy method which correspond to $\overline{T}$, T_I, T_{II} and T_h, respectively. The running time for model model (5) and the randomized greedy method do not take into account the time for computing a list of compatible sets. For this part, we set $\gamma = 5$.

Table III shows all the results for different network instances and different number of packets. The '*' in the table means that we could not obtain the result in one hour. First we observe that the comparative analysis of the delay for the four methods in Section VI-B still holds for all instances in this table. The randomized greedy method provides delays quite close to the optimum for small instances. For larger instances when the optimum is not known, it still can compute the delay which is always smaller than T_I and not far from $\overline{T}$.

Regarding the running time, we can see that the model for T_{II} is the most time consuming, and even it cannot be solved within one hour for 10-nodes network and 10 packets. Its running time increases dramatically with the number of nodes and with the number of packets. The model for T_I can be solved quickly for large networks when there are not many packets. However, it cannot be solved in one hour for 20-nodes network with 10 packets. The model for $\overline{T}$ is not impacted much by the number of packets but it's sensitive to the size of the network. Significantly, the randomized greedy method works quite fast for all cases. For 30 nodes network and 15 packets, we do not provide the results for T_h since, in this implementation, we need the list of compatible set produced by Phase 1 of Scheme I, although we can find such list of compatible sets by other ways.

TABLE III: Comparison of the delay and the running time for all methods

Networks			$\lvert\mathcal{S}\rvert = 5$		$\lvert\mathcal{S}\rvert = 10$		$\lvert\mathcal{S}\rvert = 15$	
$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$		delay	time(s)	delay	time(s)	delay	time(s)
10	31	$\overline{T}$	4.7	2	10	2	15.5	3
		T_I	7.0	15	14	855	*	*
		T_{II}	6.0	25	*	*	*	*
		T_h	6.2	1	12	2	18	5
15	75	$\overline{T}$	5.2	17	11	30	16	43
		T_I	8.0	34	18	1547	*	*
		T_{II}	6.0	967	*	*	*	*
		T_h	6.9	10	13	18	20	24
20	145	$\overline{T}$	7.1	235	14.8	282	16.7	340
		T_I	12.7	135	*	*	*	*
		T_{II}	*	*	*	*	*	*
		T_h	9.3	25	18.1	29	23.5	32
25	271	$\overline{T}$	7.7	377	15.8	442	18	503
		T_I	15.3	368	*	*	*	*
		T_{II}	*	*	*	*	*	*
		T_h	11.3	37	20.4	42	28	55
30	377	$\overline{T}$	10	2456	16.3	3045	*	*
		T_I	18.7	864	*	*	*	*
		T_{II}	*	*	*	*	*	*
		T_h	14	53	22.4	61	-	-

VII. Conclusion

In this paper we make a comprehensive study of the problem of minimizing end-to-end delay in wireless networks under physical interference model from the optimization viewpoint. We propose two schemes, where the first extends the classical minimum frame-length scheduling (Scheme I), and the second introduces a novel delay-minimization scheduling (Scheme II). Scheme I minimizes the delay retaining the high efficiency in transmission parallelization of the original minimum frame-length, while Scheme II gives an efficient instrument to fully optimize the delay. We present a set of novel formulations for the two schemes and provide theoretical analysis on their computational complexity and solution characterization. Besides, an efficient randomized greedy method is also proposed for Scheme II. The numerical results show that Scheme II can perform much better than Scheme I in terms of delay, and verify the efficiency of the heuristic method. The factors impacting the gap in delay of the two schemes are also analyzed. This work sheds some light for future study on the delay minimization problem, and the presented models can be used as benchmarks for future coming approximate and heuristic methods. The new characteristics of the models based on new sets of variables allows also to consider future extensions where advanced physical layer technologies, like interference cancellation and cooperative transmission, can be easily included.

References

[1] M. Andrews and M. Dinitz. Maximizing Capacity in Arbitrary Wireless Networks in the SINR Model: Complexity and Game Theory. In *IEEE INFOCOM*, pages 1332–1340, April 2009.

[2] O. Goussevskaia, R. Wattenhofer, M.M. Halldorsson, and E. Welzl. Capacity of Arbitrary Wireless Networks. In *IEEE INFOCOM*, pages 1872–1880, April 2009.

[3] M. Dinitz. Distributed Algorithms for Approximating Wireless Network Capacity. In *IEEE INFOCOM*, pages 1–9, March 2010.

[4] P. Bjorklund, P. Varbrand, and Di Yuan. Resource Optimization of Spatial TDMA in Ad hoc Radio Networks: a Column Generation Approach. In *IEEE INFOCOM*, pages 818–824, March 2003.

[5] P. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, and S. Tang. Wireless Link Scheduling under Physical Interference Model. In *IEEE INFOCOM*, pages 838–845, April 2011.

[6] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli. Routing, Scheduling and Channel Assignment in Wireless Mesh Networks: Optimization Models and Algorithms. *Ad Hoc Networks*, 8(6):545–563, 2010.

[7] M. Pióro, M. Żotkiewicz, B. Staehle, D. Staehle, and D.Yuan. On Maxmin Fair Flow Optimization in Wireless Mesh Networks. *Ad Hoc Networks*, 13(0):134–152, 2014.

[8] Parth H. Pathak and R. Dutta. A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks. *IEEE Communications Surveys Tutorials*, 13(3):396–428, 2011.

[9] M.F. Uddin and C. Assi. Joint Routing and Scheduling in WMNs with Variable-Width Spectrum Allocation. *IEEE Transactions on Mobile Computing*, 12(11):2178–2192, 2013.

[10] J. El-Najjar, C. Assi, and B. Jaumard. Joint Routing and Scheduling in WiMAX-Based Mesh Networks. *IEEE Transactions on Wireless Communications*, 9(7):2371–2381, 2010.

[11] L. Jun, C. Rosenberg, and A. Girard. Engineering Wireless Mesh Networks: Joint Scheduling, Routing, Power Control, and Rate Adaptation. *IEEE/ACM Transactions on Networking*, 18(5):1387–1400, 2010.

[12] W. Qing, D.O. Wu, and F. Pingyi. Delay-constrained Optimal Link Scheduling in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, 59(9):4564–4577, 2010.

[13] J. Kim, J. Cha, and H. Park. New Delay-efficient TDMA-based Distributed Schedule in Wireless Mesh Networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):369, 2012.

[14] P. Djukic and S. Valaee. Link Scheduling for Minimum Delay in Spatial Re-Use TDMA. In *IEEE INFOCOM*, pages 28–36, May 2007.

[15] A. Pantelidou and A. Ephremides. Minimum-length Scheduling with Rate Control in Wireless Networks: A Shortest Path Approach. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):115, 2011.

[16] L. Badia, A. Botta, and L. Lenzini. A Genetic Approach to Joint Routing and Link Scheduling for Wireless Mesh Networks. *Ad Hoc Networks*, 7(4):654–664, 2009.

[17] V. Gabale, A. Chiplunkar, B. Raman, and P. Dutta. DelayCheck: Scheduling Voice over Multi-hop Multi-channel Wireless Mesh Networks. In *Third International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–10, January 2011.

[18] P. Cappanera, L. Lenzini, A. Lori, G. Stea, and G. Vaglini. Optimal Joint Routing and Link Scheduling for Real-time Traffic in TDMA Wireless Mesh Networks. *Computer Networks*, 57(11):2301–2312, 2013.

[19] M. Cheng, Q. Ye, and L. Cai. Cross-Layer Schemes for Reducing Delay in Multihop Wireless Networks. *IEEE Transactions on Wireless Communications*, 12(2):928–937, 2013.

[20] A. Capone, S. Gualandi, and D. Yuan. Joint Routing and Scheduling Optimization in Arbitrary Ad Hoc Networks: Comparison of Cooperative and Hop-by-hop Forwarding. *Ad Hoc Networks*, 9(7):1256–1269, 2011.

[21] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, 2004.

[22] Laurence A. Wolsey. *Integer Programming*. Wiley Interscience, 1998.

[23] O. Goussevskaia, Y. Anne Oswald, and R. Wattenhofer. Complexity in Geometric SINR. In *ACM Mobihoc*, pages 100–109, 2007.

[24] David S Garey, Michael R.; Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[25] S.C. Huang, W. Peng-Jun, D. Jing, and Y. S Han. Broadcast scheduling in interference environment. *IEEE Transactions on Mobile Computing*, 7(11):1338–1348, 2008.

[26] GUROBI OPTIMIZAITON. http://www.gurobi.com/. Accessed: 2014-07-10.