Minimizing End-to-End Delay in Multi-Hop Wireless Networks with Optimized Transmission Scheduling

Antonio Capone[a], Yuan Li[b,*], Michał Pióro[c], Di Yuan[d]

[a] Dipartimento di Elettronica e Informazione, Politenico di Milano, Milano, Italy
[b] National Innovation Institute of Defense Technology, Beijing, China
[c] Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
[d] Department of Science and Technology, Linköping University, Norrköping,Sweden

Abstract

The problem of transmission scheduling in single hop and multi-hop wireless networks has been extensively studied. The focus has been on optimizing the efficiency of transmission parallelization, through a minimum-length schedule that meets a given set of traffic demands using the smallest possible number of time slots. Each time slot is associated with a set of transmissions that are compatible with each other according to the considered interference model. The minimum-length approach maximizes the resource reuse, but it does not ensure minimum end-to-end packet delay for multiple source-destination pairs, due to its inherent assumption of frame periodicity. In the paper we study the problem of transmission scheduling and routing aiming at minimizing the end-to-end delay under the signal-to-interference-and-noise-ratio (SINR) model for multi-hop networks. Two schemes are investigated. The first scheme departs from the conventional scheduling approach, by addressing explicitly end-to-end delay and removing the restriction of frame periodicity. The second scheme extends the first one by featuring cooperative forwarding and forward interference cancellation. We study the properties of the two schemes, and propose novel mixed-integer programming models and solution algorithms. Extensive results are provided to gain insights on how the schemes perform in end-to-end delay.

[*] Corresponding author

Email addresses: capone@elet.polimi.it (Antonio Capone), yuan.li@nudt.edu.cn (Yuan Li), mpp@tele.pw.edu.pl (Michał Pióro), di.yuan@liu.se (Di Yuan)

Preprint submitted to Ad Hoc Networks *October 14, 2018*

Keywords: Cooperative forwarding, forward interference cancellation, link scheduling, mathematical programming, SINR model, wireless networks

1. Introduction

The problem of transmission scheduling in single hop and multi-hop wireless networks has rather different characteristics depending on the considered scenarios that can range from dynamic environments, like for example mobile networks and ad hoc networks, to those with static nodes and topologies, like sensor networks of any type as well as wireless mesh networks. For the first scenarios flexibility and adaptability to changes are important characteristics of scheduling schemes, and for the second ones the efficient use of radio and energy resources is a key objective allowed by the predictability of channel and network conditions. In those cases where in addition to stable channel and topology, the network has also almost constant traffic requests, scheduling and resource optimization can be strictly controlled with centralized time division multiple access techniques. Several domains of application have these characteristics: sensor networks with large scale deployments, data gathering and processing for industrial applications, infrastructure monitoring systems, road-side networks for vehicular applications, sensors for smart agriculture, etc. As for radio technologies, the optimized scheduling approach can be supported by all those that allow time division multiple access, such as WirelessHART, ISA100, and low-rate personal area networks (LR-WPAN) based on the IEEE 802.15.4 family. Also, even if some random access technology is used, optimized scheduling remains useful as a performance benchmark.

In wireless networks the main way of maximizing the use of radio resources is through parallel transmissions from multiple nodes in such a way that the mutual interference at the receivers is sufficiently small to allow correct decoding. With a single modulation and coding scheme at the physical layer, striving for transmission parallelization is equivalent to the maximization of the overall network capacity. The so called physical interference model assumes that transmissions are successful if the signal-to-interference-and-noise ratio (SINR) at the intended receivers meets a threshold. For applications where energy efficiency is relevant, such as sensor networks, transmissions can be organized in periodic waves according to a duty cycle such that nodes are kept active only for a short scheduling period and then put to sleep for the rest of the time.

In the literature, the problem of scheduling in single hop and multi-hop wireless networks has been studied extensively, see, e.g., [1, 2, 3]. The focus so far has been on optimizing the efficiency of transmission parallelization, through a minimum-length schedule that meets a given set of traffic demands using the smallest possible number of time slots [4]. Each time slot is associated with a set of transmissions that are compatible with each other according to the interference model (compatible set). The minimum-length approach maximizes the resource reuse since all transmissions are packed into a shortest possible time frame that can then be repeated periodically.

The traffic demand is typically given as the number of packets to be transmitted for each wireless link. In multi-hop wireless networks with multiple source-destination pairs, the number of packets to be transmitted over a given link depends on the packet routing that is subject to optimization as well in the context of cross-layer optimization [5, 6]. The optimization models can also be extended to the case of rate control (multiple modulation and coding schemes at the physical layer) and power control [7].

Conventional minimum-length scheduling just ensures that a sufficient number of transmissions are scheduled for each link [8]. The sequence of compatible sets in the frame is of no significance for the optimization. However, the sequence used does matter for the end-to-end delay of a packet. In fact, the sequence may be “out of order” with respect to packet routing. If, for example, the second hop appears before the first hop in the sequence, the scheduled transmission of the former cannot be realized since the packet is not yet present for the second-hop transmission. However, because an inherent (and often implicit) assumption in minimum-length scheduling is traffic periodicity, by repeating the frame, in the long run all link transmissions scheduled within the frame will be fully utilized.

On the other hand, if the target is to deliver a set of packets without periodicity or with waves of transmissions (like in sensor networks with duty cycles), minimum-length scheduling is no longer appropriate from the delay standpoint.

Motivated by the observation above, in this paper we study the problem of minimizing the end-to-end delay of delivering a set of packets of multiple source-destinations pairs in multi-hop wireless networks. The problem is particularly relevant for applications requiring timely delivery of a set of packets, e.g., data gathering of time-critical information to be delivered to gateways and for alert message broadcasting/multicasting in delay-sensitive networks [9], for which the end-to-end delay is of concern. Moreover, the problem setting applies to scenarios with continuous traffic arrival though without periodicity. In such a context, repeatedly solving the delay minimization problem for new sets of packets forms the core module.

The end-to-end delay is defined as the total number of time slots required for delivering a given set of packets from their respective sources to destinations. By definition, our overall end-to-end delay objective coincides with the min-max delay of all packets, and thus the two terms will be used interchangeably. Our problem setup extends to joint scheduling and routing. In fact, it will become apparent later on, that scheduling and routing are inherently intertwined in the context of delay minimization.

We consider two schemes for minimizing end-to-end delay, without the constraint of frame periodicity, i.e., one single frame is constructed for the given set of packets. *Scheme I* adopts standard packet forwarding. This means that each time slot of the frame applies a compatible set, i.e., a set of simultaneous transmissions that do not interfere with each other. We prove that the delay-minimization problem for Scheme I is $\mathcal{NP}$-hard, and provide an integer programming formulation that seamlessly integrates transmission scheduling and routing, by introducing variables that naturally describe the propagation of packets (i.e., which packets are present at each node in every time slot), along with a solution algorithm that performs scheduling and routing on a slot-by-slot basis.

Next, we consider *Scheme II* that incorporates cooperative forwarding (CF) and forward interference cancellation (FIC), with the motivation of examining to what extent these two techniques can improve the delay metric. CF here refers to that multiple nodes (transmitters) can send the same packet to one node (receiver), and the receiver can combine the transmissions in order to achieve better SINR. FIC refers to the possibility when a node overhears or caches a packet, the node can cancel the interference caused by subsequent retransmissions of this packet (assuming knowledge of the scheduling solution). For Scheme II, we extend the integer programming model and the solution algorithm for Scheme I to account for the effects of CF and FIC.

We remark that, as CF amounts to activating multiple nodes for transmitting the same packet to a common receiving node, a prerequisite of CF is that the packet in question has been received by all the transmitting nodes prior to CF. Similarly, a node can perform FIC for a transmission only if it possesses the packet subject to cancelation. Clearly, whether or not these conditions are fulfilled depends on how the transmissions of packets are sequenced. Hence CF and FIC are of particular interest for the delay metric, for which the sequence of packet transmission has to be explicitly addressed, whereas extending the conventional minimum-length scheduling with CF and FIC makes much less sense.

Each of CF and FIC can be deployed alone, and, in general, which one gives better performance is scenario-specific, as will be illustrated later. Note that CF and FIC are complementary in addressing the delay, because the two have the effects of improving the nominator and denominator of SINR, respectively.

The paper is organized as follows. Section 2 provides an example to illustrate the main idea of the schemes studied in this paper. In Section 3 we review related work. Section 4 is devoted to the system model. In Section 5–6, optimization formulations, complexity analysis, and solution approaches are presented for the two delay-minimization schemes, respectively. We present numerical results in Section 7, followed by concluding remarks in Section 8.

2. An Introductory Example

It is instructive to compare, with an illustrative example, the delay incurred by the conventional minimum-length schedule with repeated frames, versus that of the two new schemes. Such an example is given in Figure 1, where five nodes are evenly spaced along a line, and the interference range induced by channel gain and SINR threshold is one hop. This means that only links that are at least two hops away are compatible. There are two packets, shown in white and black, respectively.

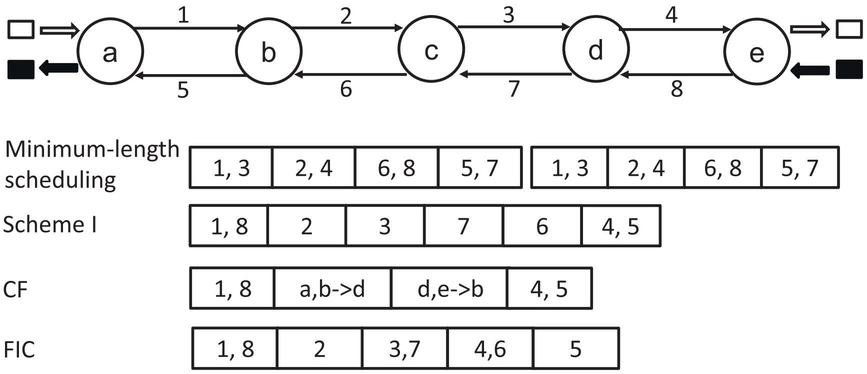


Figure 1: An illustration of the end-to-end delay.

With conventional minimum-length scheduling, the minimum frame length is four time slots with the corresponding compatible sets including the four possible pairs of links that are exactly two hops away. In order to deliver both packets, the frame must be repeated two times and then eight time slots are necessary in total for end-to-end delivery. The white packet is delivered in six slots (transmitted on links 1 and 2 in the first and second slot of the first frame, and on links 3 and 4 in the first and second slot of the second frame), while the black packet reaches its destination in eight slots (transmitted on links 8 and 7 in the third and fourth slot of the first frame, and on links 6 and 5 in the third and fourth slot of the second frame). For Scheme I, the optimum is a single frame of length six, as indicated in the figure: the first and the last compatible sets include two links, respectively links 1 and 8 and links 4 and 5, while the other compatible sets include one link only. Obviously, the frame is longer that the previous one, but it allows both packets to be delivered in six slots. For Scheme II, the effects of CF and FIC are shown separately. With CF, for the packet in white, the delay is reduced by two in comparison with the optimal delay of Scheme I: in the first slot, white packet is transmitted to node b and black packet to node d; then in the second slot nodes a, b transmit cooperatively white packet directly to d due to the extended transmission range allowed by the increased received power; similarly, in the third slot d, e transmit directly black packet to b; finally both packets make their last hop in the fourth slot. With FIC, links 3 and 7 can be active concurrently due to self-interference cancellation, and the delay is reduced by one in comparison with Scheme I.

3. Related work

The problem of maximizing the cardinality of parallel transmissions, which is a basic building block in any scheduling problem, has been well studied. In [1], the authors studied the $\mathcal{NP}$-hardness of the problem, and provided game theoretic results for power control. Approximation and distributed algorithms have been proposed in [2] and [3], respectively. For computing global optimum, an optimization method based on cutting planes is developed in [10].

For minimum-length scheduling, the amount of literature is extensive. We refer to [4] for algorithmic analysis and approximation algorithms with and without power control, and [11] for the problem version with continuous time, as well as the references therein. The work in [12] considers the feasibility version of the problem in wireless mesh networks. The extension to multiple transmission rates (obtained via multiple modulation and coding schemes) and power control has been analyzed in [7]. Joint optimization of routing and scheduling for throughput-related objectives has been analyzed under various assumptions and models, see [5]. From the modeling perspectives, the notion of compatible set, originally developed in [8], has been a mainstream method for problem formulation and solution under the SINR model. The study of fairness in wireless mesh networks [13, 14] also develops optimization models using the notion of compatible sets.

Delay minimization in multi-hop wireless networks is a current topic with a growing amount of research effort. Some references, such as [15, 16, 17], analyze the delay using queuing theory, assuming that the packet arrival process at each source is a stationary and ergodic Markov chain. The work in [18] provides analysis of end-to-end delay, by assuming that flows differ in priority and the scheduling solution follows the priority order. The authors of [19] present integer programming models for minimizing the end-to-end delay for general multi-hop wireless networks. All these works assume that the packet routes are given, and the compatible link set in each time slot is obtained either based on the conflict graph (a.k.a. the protocol model) or in a greedy manner. The conflict graph, however, is less accurate than the SINR model, because the cumulative effect of interference is not accounted for.

Some authors have studied delay with very specific problem setups. For example, the studies in [20, 21] consider delay in data gathering and broadcasting in wireless networks, respectively, with the restriction of a tree topology. In [22], the problem of finding a delay-optimal link scheduling solution under the conflict graph model is studied, assuming given link bandwidths, frame length, and routing paths. Because frame length, routing, as well as delay are inherently intertwined, our work is a generalization of the problem setting in [22], along with a more accurate model for characterizing interference.

CF and interference cancellation have shown significant benefits for improving resource efficiency. In [23], the authors define the concept of CF, which is modeled via a cooperation graph, and consider minimum-length scheduling with CF for multi-hop wireless networks with multiple sources and destinations. Overview, mechanisms and implementation details of CF can be found in [24, 25]. Interference cancellation is a technique by which a receiver decodes an interfering signal and thereby removes it from the composite signal. In this paper, we focus on forward interference cancellation (FIC). In FIC, a node can cancel the interference caused by the transmission of a packet that the node has correctly received earlier (either because the node is an intended receiver or for the purpose of FIC) and then buffered. In [26], two schemes have been presented for implementing FIC for fast and low fading channels respectively. Improvement in throughput by FIC has been analyzed via simulation in [27]. The authors of [28] show the benefit of FIC for a two-user scenario in cognitive radio networks.

4. Network model

4.1. General setup and notations

A wireless network is modeled by means of a bi-directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of bidirectional links, and $\mathcal{A} = \{(i, j), (j, i) : \{i, j\} \in \mathcal{E}\}$ is the set of directed links (i.e., arcs). We assume that $\{i, j\} \in \mathcal{E}$ if, and only if, the signal-to-noise ratio (SNR) condition is satisfied, i.e., $\frac{p(i,j)}{\eta} \geq \gamma$ and $\frac{p(j,i)}{\eta} \geq \gamma$, where $p(i, j)$ and $p(j, i)$ are the received power at node j when node i is transmitting and vice versa, respectively, defined as $p(i, j) := P(i)g(i, j)$, $p(j, i) := P(j)g(j, i)$. Here, $P(i)$ and $P(j)$ are the transmission powers of nodes i and j, respectively, $g(i, j) = g(j, i)$ is the gain of link $\{i, j\}$, η is the noise power, and γ is the given SNR threshold. The set of nodes incident to $i \in \mathcal{V}$ (neighbors of i) is denoted by $\Gamma(i)$, i.e., $\Gamma(i) = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$.

Consider a sequence of time slots $\mathcal{T} := \{1, 2, \ldots, T\}$. Suppose $\mathcal{W}$ is the set of nodes transmitting in a time slot. Then arc $(i, j) \in \mathcal{A}$ can be active in this time slot only if $i \in \mathcal{W}, j \notin \mathcal{W}$ and the following SINR condition is fulfilled:

$$\frac{p(i, j)}{\eta + \sum_{k \in \mathcal{W} \setminus \{i\}} p(k, j)} \geq \gamma. \tag{1}$$

We assume that at any time slot a node can either transmit, receive or be inactive (this in particular implies half-duplex transmission: the two arcs composing a bidirectional link cannot be active simultaneously in both directions). Besides, in the standard transmission model without CF or FIC, a node can transmit over at most one arc in a time slot; similarly, a node can receive over at most one arc in a time slot.

A set $c \subseteq \mathcal{A}$ of arcs that can be simultaneously active with respect to the above conditions is called a *compatible set*. In the sequel, the set of active nodes in a compatible set c is denoted by $\mathcal{W}(c) := \{i : \exists j, (i,j) \in c\}$. Thus the SINR condition applies to every arc $(i,j) \in c$:

$$\frac{p(i,j)}{\eta + \sum_{k \in \mathcal{W}(c) \setminus \{i\}} p(k,j)} \geq \gamma. \tag{2}$$

We consider a given set of packets $\mathcal{S}$. Each packet $s \in \mathcal{S}$ requires one time-slot to be transmitted over an arc, and is to be sent from its originating node $o(s)$ to its destination node $d(s)$ (in general visiting several transit nodes on its way). The packets are present at their respective origins just before the first time slot $t = 1$ starts. The time slot $t = \overline{\mathcal{T}}$ during which end-to-end delivery is accomplished for all packets determines the overall delay.

Notations used in this paper are summarized in Table 1.

4.2. Cooperative Forwarding (CF)

The idea of cooperative packet forwarding has been studied in recent years. CF dates back to the work of [29, 30]. With CF, a packet can be simultaneously forwarded by a group of nodes acting together as a virtual array of antennas. That is, the same packet can be transmitted simultaneously from several nodes to one or more receivers, and each receiver can combine these packet transmissions. Note that CF tends to improve the numerator of the SINR. In this paper, we adopt the CF scheme described in [23], in which the SINR condition for node $j \in \mathcal{V}$ to correctly receives packet $s \in \mathcal{S}$ is as follows:

$$\frac{\sum_{i \in \mathcal{W}(s)} p(i,j)}{\eta + \sum_{k \in \mathcal{W} \setminus \mathcal{W}(s)} p(k,j)} \geq \gamma. \tag{3}$$

Here, as before, $\mathcal{W}$ is a set of concurrently transmitting nodes and $\mathcal{W}(s) \subseteq \mathcal{W}$ is the subset of nodes transmitting packet s.

Table 1: Notations

Notation	Description
$\mathcal{V}$	set of nodes
$\mathcal{E}$	set of bidirectional links
$\mathcal{A}$	set of directed links (i.e., arcs)
$\Gamma(i)$	set of neighbor nodes of node i
$P(i)$	transmitting power of node i
$g(i,j)$	gain of link $\{i,j\}$
$p(i,j)$	power received at node j from transmitting node i
η	noise power
γ	SINR threshold
c	compatible set ($c \subseteq \mathcal{A}$)
$\mathcal{S}$	set of packets
$o(s), d(s)$	origin and destination, respectively, of packet $s \in \mathcal{S}$
T	an upper bound on delay (in the number of time slots) for delivering all packets in $\mathcal{S}$
$\mathcal{T}$	set of consecutive time slots ($\mathcal{T} := \{1, 2, \dots, T\}$)
D_I	delay of Scheme I
D_{II}	delay of Scheme II

Observe that with CF, the nodes store the received packets so they can transmit them more than once. Also, the notion of transmission over a single arc, as assumed in the standard packet forwarding model, is not appropriate anymore, because now the broadcasting nature of radio transmission is exploited, i.e., it becomes advantageous that all nodes for which condition (3) is met receive and store packet s.

In modeling CF, the notion of a compatible set becomes more complicated than that in the standard model in Section 4.1. Now such a set is characterized by the list of transmitting nodes (together with the packet each such node transmits), and the list of receiving nodes (together with the packet each such node receives), along with the requirement that the SINR condition (3) is fulfilled for each receiving node.

4.3. Forward Interference Cancellation (FIC)

In multi-hop wireless networks, it is common that a node receives or overhears a packet several times. The FIC technique, aimed at addressing interference due to repeated transmissions of the same packet, has been proposed in [27]. In FIC, a node can cancel the interference caused by the transmission of a packet, if the node could earlier decode (and then cache) the packet. As discussed in [27], FIC is different from the widely studied successive interference cancellation (SIC) scheme [31]. In SIC, cancellation is possible only if the interfering signal is strong enough so that the receiver can decode it. With FIC, the cancellation of an interfering transmission is not restricted by the signal strength. Moreover, FIC utilizes the fact that a receiver may have a packet from earlier (intended or interfering) transmissions, when this packet is transmitted again and causes interference. The work of [27] also presents implementation aspects of FIC. Note that an FIC-enabled node can perform self-interference cancellation, that is, the node can receive packets while transmitting. Compared to some other methods, such as analog network coding (ANC) [32], ZigZag [33] and Chorus [34], FIC can be viewed as a generic approach with broad applicability.

With FIC, the SINR condition for node $j \in \mathcal{V}$ to correctly receive packet $s \in \mathcal{S}$ from node i is as follows:

$$\frac{p(i,j)}{\eta + \sum_{k \in \mathcal{W} \setminus (W(j) \cup \{i\})} p(k,j)} \geq \gamma, \tag{4}$$

where $\mathcal{W}(j) \subseteq \mathcal{W}$ is the subset of nodes transmitting packets that have been stored in node j.

4.4. The minimum delay scheduling problem

The optimization problem considered in this paper is referred to as the minimum delay scheduling problem (MDSP). It consists in minimizing the overall end-to-end delay through joint optimization of packet transmission scheduling (along the time slots) and packet routing. In essence, MDSP determines, for each time slot, the subset of packets to be transmitted and the links to be used for the transmissions, with the objective of using a minimum number of time slots to deliver all the packets to their respective destinations. In the two subsequent sections we will study two versions of MDSP, with standard packet forwarding (S-MDSP) and the extension with CF/FIC packet forwarding (E-MDSP), respectively, using exact integer programming models as well as heuristic methods.

5. Scheme I: minimum delay scheduling with standard packet forwarding

In this section we consider Scheme I, and study the resulting optimization problem S-MDSP.

5.1. Formulation of S-MDSP

Let T denote an upper bound on the delay, with $\mathcal{T} = \{1, 2, \ldots, T\}$. In order to set up a mathematical model, we need to determine or keep track on, for each time slot, which packets are present at which nodes, and which nodes should transmit and receive which packets, if the time slot is used at all. We will use the following binary variables: λ^t – equal to 1 if slot $t \in \mathcal{T}$ is actually used; X^t_{is} – equal to 1 if node $i \in \mathcal{V}$ is sending $s \in \mathcal{S}$ in $t \in \mathcal{T}$; Y^t_{is} – equal to 1 if $i \in \mathcal{V}$ is receiving $s \in \mathcal{S}$ in $t \in \mathcal{T}$; y^t_{is} – equal to 1 if $s \in \mathcal{S}$ is present at $i \in \mathcal{V}$ by $t \in \mathcal{T}$. S-MDSP can be formulated as follows.

In the formulation, objective (5a) minimizes the total number of slots required to deliver all packets from their origins to destinations, i.e., the delay. Constraint (5b) forces all the slots after the first unused one to be unused as well (then the actual frame length is equal to $\max\{t \in \mathcal{T} : \lambda^t = 1\}$). Constraint (5c) ensures that, in any slot, a node can either transmit or receive a packet, or do nothing. Constraint (5d) states that in any slot, a packet can be transmitted by at most one node and received by at most one (other) node.

$$\text{minimize } \; D_I = \textstyle\sum_{t \in \mathcal{T}} \lambda^t \tag{5a}$$

$$\lambda^{t+1} \leq \lambda^t, \; t \in \mathcal{T} \setminus \{T\} \tag{5b}$$

$$\textstyle\sum_{s \in \mathcal{S}} (X_{is}^t + Y_{is}^t) \leq \lambda^t, \; i \in \mathcal{V}, t \in \mathcal{T} \tag{5c}$$

$$\textstyle\sum_{i \in \mathcal{V}} X_{is}^t \leq 1, \; \sum_{i \in \mathcal{V}} Y_{is}^t \leq 1, \; s \in \mathcal{S}, t \in \mathcal{T} \tag{5d}$$

$$\textstyle\sum_{i \in \Gamma(j)} X_{is}^t \geq Y_{js}^t, \; j \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T} \tag{5e}$$

$$\textstyle\sum_{t \in \mathcal{T}} X_{is}^t \leq 1, \; \sum_{t \in \mathcal{T}} Y_{is}^t \leq 1, \; i \in \mathcal{V}, s \in \mathcal{S} \tag{5f}$$

$$y_{is}^t \leq \textstyle\sum_{\tau=1}^{t} Y_{is}^{\tau}, \; s \in \mathcal{S}, i \in \mathcal{V} \setminus \{o(s)\}, t \in \mathcal{T} \tag{5g}$$

$$y_{o(s)s}^1 = 1, \; y_{d(s)s}^T = 1, \; s \in \mathcal{S} \tag{5h}$$

$$X_{is}^t \leq y_{is}^t, \; i \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T} \tag{5i}$$

$$\begin{aligned} & p(i,j) + M(j)(1 - X_{is}^t) + M(j)(1 - Y_{js}^t) \geq \\ & \qquad \geq \gamma(\eta + \textstyle\sum_{k \in \mathcal{V} \setminus \{i,j\}} \sum_{r \in \mathcal{S} \setminus \{s\}} p(k,j) X_{kr}^t), \; (i,j) \in \mathcal{A}, s \in \mathcal{S}, t \in \mathcal{T} \end{aligned} \tag{5j}$$

$$\lambda, X, y, Y \; \text{ binary.} \tag{5k}$$

Constraint (5e) makes sure that each node j can receive packet s in slot t only if one of its neighbors is transmitting s in this slot. Inequality (5f) permits a node to send, as well as to receive, any particular packet at most once during the frame. This constraint, as well as constraint (5d), is justified because it is sufficient to transmit the packet only along its routing path in the hop-by-hop manner. Constraint (5g) and (5h) define variables y and set conditions on them for the beginning and the end of the schedule. Constraint (5i) prohibits a packet from being transmitted by a node before it has been received. Finally, constraint (5j) expresses the SINR condition for transmitting packet s on arc (i, j) in slot t. Since the SINR requirement does matter only when $X_{is}^t = Y_{js}^t = 1$ (i.e., when s is transmitted over arc (i, j)), the “big M” value $M(j)$ is used in the left-hand side of (5j) to cancel this requirement whenever $X_{is}^t \cdot Y_{js}^t = 0$. For that $M(j)$ can for example be set to $\gamma(\eta + \sum_{k \in \mathcal{N} \setminus \{j\}} p(k, j))$ – this corresponds to the case when all nodes besides j are transmitting. However, when j receives packet s from node i in slot t (i.e., when $X_{is}^t = Y_{js}^t = 1$), then (5j) becomes equivalent to

$$p(i,j) \geq \gamma(\eta + \textstyle\sum_{k \in \mathcal{V} \setminus \{i,j\}} \sum_{r \in \mathcal{S} \setminus \{s\}} p(k,j) X_{kr}^t), \tag{6}$$

which precisely ensures that the SINR threshold (2) is met. Note that the second summation (over $\mathcal{S} \setminus \{s\}$) on the right-hand side of (6) is valid since when $X_{is}^t = 1$ then all nodes besides i are forbidden to transmit s in t because of (5d).

Observe that the formulation of S-MDSP can be extended to consider the case where packets can be added to set $\mathcal{S}$ at a later time instant, say, at the beginning of time slot $\hat{t} + 1$, when the optimized scheduling is already being executed. It is easy to see that the rest of the scheduling (from time slot $\hat{t} + 1$ on) can be re-optimized using virtually the same formulation of S-MDSP.

Model (5) uses $O(|\mathcal{V}||\mathcal{S}||\mathcal{T}|)$ variables and $O(|\mathcal{A}||\mathcal{S}||\mathcal{T}|)$ constraints. The size is compact, in order to represent the presence of packets at nodes as well as the transmissions of packets along the time line. Also, we remark that this type of modeling has not been considered for the end-to-end delay in the current literature.

5.2. Complexity of S-MDSP

Theorem 1. *S-MDSP is $\mathcal{NP}$-hard even for the single-hop networks.*

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 becomes to minimum-length scheduling, which is NP-hard [8, 11]. □

Each feasible solution of the S-MDSP formulation (5) specifies, for each time slot $t \leq D_I$, the compatible set of links $c(t) := \{(i,j) \in \mathcal{A} : \exists s \in \mathcal{S}, X_{is}^t \cdot Y_{js}^t = 1\}$. Thus, in formulation (5) the compatible sets used in the consecutive slots of the frame are subject to optimization. Certainly, the delay minimization problem like S-MDSP arises also in the scenarios where the list of compatible sets to be used in the frame is given. Then the compatible sets are not optimized, only their assignment to the frame slots. The complexity of this problem is provided in Theorem 2. The proof of this theorem is more tedious to demonstrate and thus has been moved to the appendix.

Theorem 2. *S-MDSP with given compatible sets is $\mathcal{NP}$-hard.*

5.3. A heuristic algorithm for S-MDSP

Table 2: Notations for the heuristic.

Notation	Description
$\mathcal{S}(t)$	set of packets not yet delivered to their respective destinations by the beginning of slot t
$\mathcal{S}(i,t)$	set of packets from $\mathcal{S}(t)$ present at node $i \in \mathcal{V}$
$\mathcal{S}'(i,t)$	$:= \mathcal{S}(t) \backslash \mathcal{S}(i,t)$, set of packets from $\mathcal{S}(t)$ not present at node $i \in \mathcal{V}$
$\mathcal{V}(s,t)$	set of nodes having packet $s \in \mathcal{S}$ at the beginning of slot t
$\mathcal{V}'(s,t)$	$:= \mathcal{V} \setminus \mathcal{V}(s,t):$ set of nodes not having packet $s \in \mathcal{S}$ at the beginning of slot t
$l(s,i)$	distance (minimum number of hops) from node $i \in \mathcal{V}$ to $d(s)$
$L(s,t)$	$:= \min_{i \in \mathcal{V}(s,t)} l(s,i)$, current distance of packet $s \in \mathcal{S}(t)$ to its destination $d(s)$.
X_{is}	binary variable that is equal to 1 if node $i \in \mathcal{V}(s,t)$ is transmitting packet $s \in \mathcal{S}(t)$, and 0 otherwise
Y_{js}	binary variable that is equal to 1 if node $j \in \mathcal{V}'(s,t)$ is receiving packet $s \in \mathcal{S}(t)$, and 0 otherwise
Z_{js}	binary variable that is equal to 1 if node $j \in \mathcal{V}'(s,t)$ is the closest node to $d(s)$, reached by packet $s \in \mathcal{S}(t)$ after the transmission
Z_{0s}	auxiliary binary variable for $s \in \mathcal{S}(t)$
z_s	variable expressing the distance to destination of packet $s \in \mathcal{S}(t)$ after the transmissions in slot t

Since S-MDSP is $\mathcal{NP}$-hard, solving (5) to optimality can be excessively time consuming. For this reason, we propose a greedy heuristic algorithm for delay minimization for S-MDSP. The algorithm schedules packet transmissions in consecutive steps corresponding to time slots. At each step, it minimizes, in a greedy way, an objective function related to minimization of the maximum delay over the packets not yet delivered to their destinations. The notation used by the algorithm is summarized in Table 2, and the method is described in Algorithm 1.

Step 0 of Algorithm 1 provides the initial status of all packets. In Step 1 mixed-integer programming (MIP) formulation (7) – a single-slot version of (5) is used. In (7) constraints (5d) and (5f) are skipped. This modification does not alter the S-MDSP optimal solutions and at the same time makes (5) applicable for broadcast scenarios where packets have multiple destinations. Deleting the first part of (5d) enables several nodes to send the same packet in a time slot and deleting the second part of (5d) enables multiple neighbors to receive a particular packet simultaneously as long as the SINR constraint is satisfied for the corresponding transmissions.

The objective of model (7) is to minimize the total distance of all packets to their destinations after transmission in time slot t. Constraint (7b) assures that a node can either send a packet or receive a packet in time slot t. Constraint (7c) makes sure that a packet can only be sent by one node in time slot t. Constraint (7d) assures that if $Z_{js} = 0$, then $Y_{js} = 0$, which means node j should not receive packet s. Constraint (7e) decides which node can receive packet s or no nodes can receive packet s, i.e., $Z_{0s} = 0$. Next constraint (7f) computes the minimal distance of packet s from all nodes having packet s to its destination $d(s)$. Constraint (7g) expresses the SINR condition for delivering packet s from node i to node j.

Equations in Step 3 update the related sets. After transmissions in time slot t, the nodes receiving packet s ($Y^*_{js} = 1$) are added to $\mathcal{V}(s, t+1)$, see (8a). Equation (8b) records the packets that have not been delivered to their destinations in time slot $t + 1$. Equation (8c) expresses the set of packets contained in each node. Equation (8d) computes for each packet the minimal distance to its destination.

Algorithm 1 A heuristic for S-MDSP

Step 0: Put $\mathcal{S}(1) := \mathcal{S}$ (as no packet is delivered yet), $\mathcal{S}(i,1) := \{s \in \mathcal{S} : o(s) = i\}, i \in \mathcal{V}$, $\mathcal{V}(s,1) := \{o(s)\}$, $L(s,1) = l(s,o(s))$, $t := 1$.

Step 1: Solve the mixed-integer programming formulation:

$$\text{minimize } \textstyle\sum_{s\in\mathcal{S}(t)} z_s \tag{7a}$$

$$\textstyle\sum_{s\in\mathcal{S}(i,t)} X_{is} + \sum_{s\in\mathcal{S}'(i,t)} Y_{is} \leq 1,\ i \in \mathcal{V} \tag{7b}$$

$$\textstyle\sum_{i\in\mathcal{V}(s,t)} X_{is} \leq 1,\ s \in \mathcal{S}(t) \tag{7c}$$

$$Z_{js} \leq Y_{js},\ s \in \mathcal{S}(t), j \in \mathcal{V}'(s,t) \tag{7d}$$

$$Z_{0s} + \textstyle\sum_{j\in\mathcal{V}'(s,t)} Z_{js} = 1,\ s \in \mathcal{S}(t) \tag{7e}$$

$$z_s = L(s,t)Z_{0s} + \textstyle\sum_{j\in\mathcal{V}'(s,t)} l(s,j)Z_{js},\ s \in \mathcal{S}(t) \tag{7f}$$

$$\begin{aligned}&\textstyle\sum_{i\in\mathcal{V}(s,t)} p(i,j)X_{is} + M(j)(1-Y_{js}) \geq \\ &\quad \geq \gamma(\eta + \textstyle\sum_{k\in\mathcal{V}\setminus\{j\}} \sum_{r\in\mathcal{S}(t)\setminus\{s\}} p(k,j)X_{kr}), \\ &\qquad\qquad j \in \mathcal{V}, s \in \mathcal{S}'(j,t)\end{aligned} \tag{7g}$$

$$X, Y, Z \text{ binary}; z \text{ continuous.} \tag{7h}$$

Step 2: For the optimal solution X^*, Y^*, Z^*, z^* obtained in **Step 1** put:

$$\mathcal{V}(s,t+1) := \mathcal{V}(s,t) \cup \{j \in \mathcal{V}'(s,t) :\ Y^*_{js} = 1\}, s \in \mathcal{S}(t) \tag{8a}$$

$$\mathcal{S}(t+1) := \{s \in \mathcal{S}(t) :\ d(s) \notin \mathcal{V}(s,t+1)\} \tag{8b}$$

$$\mathcal{S}(i,t+1) := \{s \in \mathcal{S}(t+1) :\ i \in \mathcal{V}(s,t+1)\},\ i \in \mathcal{V} \tag{8c}$$

$$L(s,t+1) := \min_{i\in\mathcal{V}(s,t+1)} l(s,i)\ ,\ s \in \mathcal{S}(t+1). \tag{8d}$$

Step 3: If $\mathcal{S}(t+1) = \emptyset$ stop: all packets have been delivered in t slots. Otherwise, set $t := t+1$ and goto **Step 1**.

6. Scheme II: minimum delay scheduling with CF and FIC

In this section we consider Scheme II for minimum delay scheduling that assumes CF and FIC techniques. Recall that the resulting problem of minimizing the delay is denoted by E-MDSP.

6.1. Formulation of E-MDSP

The MIP formulation of E-MDSP specified in (9) uses the same variables as the S-MDSP formulation (5) plus auxiliary variables z (defined later).

$$\text{minimize } D_{II} = \textstyle\sum_{t\in\mathcal{T}} \lambda_t \tag{9a}$$

$$\lambda^{t+1} \leq \lambda^t,\ t \in \mathcal{T} \setminus \{T\} \tag{9b}$$

$$\textstyle\sum_{s\in\mathcal{S}}(X_{is}^t + Y_{is}^t) \leq \lambda^t,\ i \in \mathcal{V}, t \in \mathcal{T} \tag{9c}$$

$$\textstyle\sum_{i\in\mathcal{V}} Y_{is}^t \leq 1,\ s \in \mathcal{S}, t \in \mathcal{T} \tag{9d}$$

$$\textstyle\sum_{t\in\mathcal{T}} Y_{is}^t \leq 1,\ i \in \mathcal{V}, s \in \mathcal{S} \tag{9e}$$

$$y_{is}^t \leq \textstyle\sum_{\tau=1}^t Y_{is}^\tau,\ s \in \mathcal{S}, i \in \mathcal{V} \setminus \{o(s)\}, t \in \mathcal{T} \tag{9f}$$

$$y_{o(s)s}^1 = 1,\ y_{d(s)s}^T = 1,\ s \in \mathcal{S} \tag{9g}$$

$$X_{is}^t \leq y_{is}^t,\ i \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T} \tag{9h}$$

$$X_{is}^t \leq 1 - y_{d(s)s}^t,\ i \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T} \tag{9i}$$

$$\begin{aligned} &\textstyle\sum_{i\in\mathcal{V}\setminus\{j\}} p(i,j)X_{is}^t + M(j)(1 - Y_{js}^t) \geq \\ &\qquad \geq \gamma(\eta + \textstyle\sum_{k\in\mathcal{V}\setminus\{j\}} \sum_{r\in\mathcal{S}\setminus\{s\}} p(k,j)z_{kjr}^t),\ j \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T} \end{aligned} \tag{9j}$$

$$\begin{aligned} &z_{kjr}^t \leq X_{kr}^t,\ z_{kjr}^t \leq 1 - y_{jr}^t,\ z_{kjr}^t \geq X_{kr}^t - y_{jr}^t,\ z_{kjr}^t \geq 0, \\ &\qquad\qquad j \in \mathcal{V}, k \in \mathcal{V} \setminus \{j\}, r \in \mathcal{S}, t \in \mathcal{T} \end{aligned} \tag{9k}$$

$$\lambda, X, Y, y \text{ binary; } z \text{ continuous.} \tag{9l}$$

Objective (9a) and constraints (9b), (9c) remain unchanged with respect to formulation (5). Constraint (9d) results from (5d) by deleting the condition forbidding multiple transmissions of a packet in the slot. Constraint (5e) is skipped as it is implicitly included in the new SINR constraint (9j). Constraint (9e) is a modification of (5f) – now a packet can be sent from the same node in multiple time slots. Constraints (9f)-(9h) are maintained with additional condition (9i) explicitly forbidding transmissions of a packet after it has reached its destination.

The SINR constraint (incorporating (3) and (4)) is substantially different than its S-MDSP counterpart (5j). This is because now the notions of the transmission link and the packet route are not used as they become somewhat “fuzzy” due to the use of CF and FIC. The SINR constraint is formulated in (9j) using auxiliary variables z to get rid of bi-linearities that would otherwise appear in the right-hand side of (9j). Each such variable, z_{kjr}^t, expresses the product $X_{kr}^t \cdot (1 - y_{jr}^t)$, as forced by (9k). That is, $z_{kjr}^t = 1$ if node k is sending packet r in time slot t, and this packet is not present at node j (otherwise $z_{kjr}^t = 0$). The first term in (9j) expresses the basic property of CF: the power received at node j is summed up over all the nodes sending packet s. In the right-hand side of (9j), the transmissions carrying packets other than s but already present at node j are cancelled by FIC. As for S-MDSP, the value of the “big M” constant $M(j)$ can be set to $\gamma(\eta + \sum_{k \in \mathcal{N} \setminus \{j\}} p(k,j))$ – this corresponds to the case when all nodes (besides j) are sending packets (different than s) that have not been received at node j yet. Model (9) is of the same size as (5) in the number of constraints, whereas the number of variables is scaled up with the number of nodes. The latter is necessary in order to address CF and FIC.

6.2. Complexity of E-MDSP

Theorem 3. *E-MDSP is $\mathcal{NP}$-hard.*

Proof. For single-hop networks, CF and FIC are no use for reducing the delay of transmitting packets since each packet needs only one transmission. Then E-MDSP becomes S-MDSP and hence E-MDSP is $\mathcal{NP}$-hard. □

6.3. A heuristic algorithm for E-MDSP

The heuristic algorithm (referred to as Algorithm 2 in the sequel) for E-MDSP works essentially in the same manner as the heuristic for S-MDSP, i.e., Algorithm 1 described in Section 5.3. The difference lies in the optimization problem solved in Step 1 of Algorithm 1, as specified in (10) (using notations of Table 2). The problem is a one-slot counterpart of E-MDSP (9).

$$\text{minimize } \textstyle\sum_{s\in\mathcal{S}(t)} z_s - \varepsilon \sum_{s\in\mathcal{S}(t)} \sum_{i\in\mathcal{V}} Y_{is} \tag{10a}$$

$$(7\text{b}), (7\text{d}), (7\text{e}), (7\text{f})$$

$$\begin{aligned} &\textstyle\sum_{i\in\mathcal{V}(s)} p(i,j)X_{is} + M(j)(1-Y_{js}) \geq \\ &\qquad \geq \gamma(\eta + \textstyle\sum_{r\in\mathcal{S}'(j,t)\setminus\{s\}} \sum_{k\in\mathcal{V}(r,t)} p(k,j)X_{kr}),\ j\in\mathcal{V}, s\in\mathcal{S}'(j,t) \end{aligned} \tag{10b}$$

$$X, Y, Z \text{ binary}, z \text{ continuous}. \tag{10c}$$

In the above, ε is a positive constant used in the second term of objective (10a) to express the secondary optimization goal, that is, maximizing the number of transmissions. The reason of introducing this secondary goal is that in general the more transmissions in the time slot the more packets delivered to the neighboring nodes. This potentially enables increase in the use of CF and FIC in the subsequent time slots, which is helpful in reducing packet delay.

Note that constraint (7c) is not included in model (10), which makes it possible that a packet can be simultaneously sent by several nodes. With (10b), CF and FIC can also work in each time slot.

7. Numerical Study

Below we report the results of a numerical study that examines the optimization models described in the previous sections, and also compares the performance of the two schemes. In all experiments, the noise power is $\eta = 10^{-13}$ W, and the transmission power for all nodes is $P = 0.1$ W. The power gain between node i and node j is $g_{ij} = d_{ij}^{-4}$ where d_{ij} is the distance. Given SINR threshold γ, the maximal transmission distance can be computed as $L = (P/(\gamma \times \eta))^{\frac{1}{4}}$, which is derived from the SNR condition. All optimization formulations have been solved by the Gurobi solver on an Intel i7-5600U at 2.6 GHz, with 8 GB RAM.

As a comparison, we introduce the approach described in [35], "solution driven by minimum-frame-length scheduling", which minimizes the end-to-end

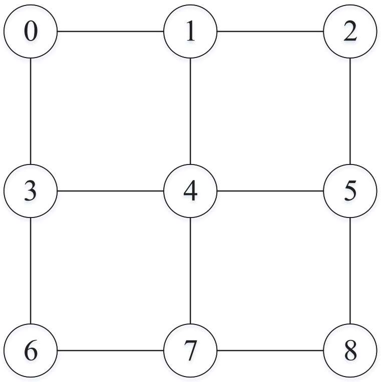


Figure 2: A grid network example.

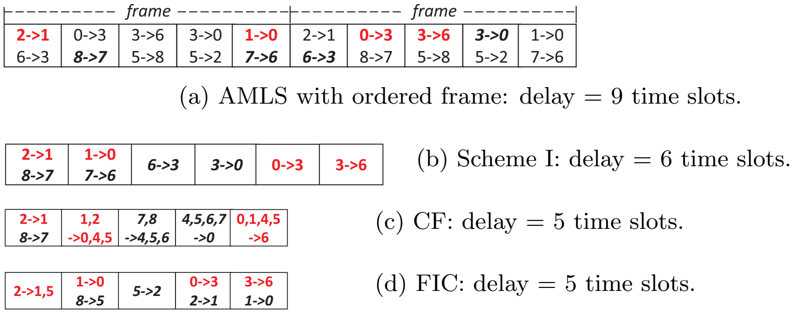
delay based on repeated frames. The approach includes two phases. The first phase is to find the minimum frame length (the number of compatible sets) by minimum-length scheduling such that sufficient links are scheduled for all node-pairs. The second phase is to find the optimal order of compatible sets in the frame with the objective of the delay. We define this approach as "AMLS"(approach of minimum-length scheduling).

7.1. An illustrative example

We use a grid network depicted in Figure 2 as an example to illustrate the optimal solutions and the advantages in reducing the delay of the two schemes. The length of the edge of the grid network is 250 m. We consider two packets (node-pairs): from node 2 to node 6 ($s = 1$) and from node 8 to node 0 ($s = 2$), and the SINR threshold is set to $\gamma = 10$.

Figure 3 shows scheduling solutions of AMLS, Scheme I and Scheme II. Each square corresponds to a time slot and indicates the transmitting links. Packet $s = 1$ is transmitted along red bold links while packet $s = 2$ is transmitted along black italic links. With minimum-length scheduling, we find a frame composed of a list of ordered compatible sets, shown as the frame in Figure (3a).The transmissions along the paths required for delivering the two packets, i.e., $2 \rightarrow 1 \rightarrow 0 \rightarrow 3 \rightarrow 6$ and $8 \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow 0$, are included in the frame.

With this frame, the two packets are delivered to their destinations in 9 time slots and the frame is repeated twice.



frame					frame				
2->1 6->3	0->3 8->7	3->6 5->8	3->0 5->2	1->0 7->6	2->1 6->3	0->3 8->7	3->6 5->8	3->0 5->2	1->0 7->6

(a) AMLS with ordered frame: delay = 9 time slots.

2->1 8->7	1->0 7->6	6->3	3->0	0->3	3->6

(b) Scheme I: delay = 6 time slots.

2->1 8->7	1,2 ->0,4,5	7,8 ->4,5,6	4,5,6,7 ->0	0,1,4,5 ->6

(c) CF: delay = 5 time slots.

2->1,5	1->0 8->5	5->2	0->3 2->1	3->6 1->0

(d) FIC: delay = 5 time slots.

Figure 3: The frames optimized for different assumptions.

AMLS mainly improves transmission parallelization, i.e., parallelizing as many transmissions as possible, which is not optimal from the delay standpoint. For Scheme I, the delay is addressed directly, that is, one single frame is constructed for the given packets with the delay objective. As we can see in Figure (3b), the delay is 6 time slots, which saves 3 time slots compared to Figure (3a). With CF, the delay is further reduced by 1, see Figure (3c). The CF is used in time slots $2, 3, 5$, in which a set of nodes (listed on the left-hand side of the arrow) simultaneously send a packet to revivers (listed on the right-hand side of the arrow). Take transmissions in time slot 5 for example, nodes $0, 1, 4, 5$ can send packet $s = 1$ to node 6 directly, since $\frac{p(0,6)}{\eta} = 1.6 < \gamma$, $\frac{p(1,6)}{\eta} = 1.024 < \gamma, \frac{p(4,6)}{\eta} = 6.4 < \gamma$ and $\frac{p(5,6)}{\eta} = 1.024 < \gamma$. But by CF of nodes $0, 1, 4, 5$, $\frac{p(0,6)+p(1,6)+p(4,6)+p(5,6)}{\eta} = 10.048 > \gamma$. With FIC, the delay is also 5 slots which saves 1 slot compared to Scheme I. In time slot 4 of Figure (3d), node 1 can successfully receive packet $s = 2$ from node 2 even if node 0 is transmitting packet $s = 1$. The interference brought by node 0 is cancelled at node 1 since node 1 already buffers packet $s = 1$.

7.2. Comparison of two schemes

In the following we make extensive numerical studies to compare the two schemes. The test scenarios are generated by randomly distributing nodes in a square area of 1000 m × 1000 m. The SINR threshold is set as $\gamma = 10$.

A. Computational efficiency

We present the computational efficiency for solving the optimization models of AMLS, Scheme I, Scheme II and the two heuristics (Heuristic I and Heuristic II) corresponding to the two schemes. The computing time is shown in Table 3. Notation '*' in the table means that the solution cannot be obtained within a time limit of one hour. We test networks with different number of nodes. For each network instance, packets are randomly selected among all node-pairs.

Table 3: A comparison of delay and computing time.

$\|\mathcal{V}\|$	$\|\mathcal{S}\|$	AMLS		Scheme I		Scheme II		Heuristic I		Heuristic II	
		delay	time(s)	delay	time(s)	delay	time(s)	delay	time(s)	delay	time(s)
15	2	2.5	6s	2.3	2s	2.0	3s	2.5	< 1s	2.4	< 1s
	4	8.1	24s	6.3	568s	5.3	810s	8.3	< 1s	6.7	< 1s
	6	10.5	50s	*	*	*	*	11.0	< 1s	9.7	< 1s
25	2	6.7	716s	5.3	25s	4.7	58s	5.7	< 1s	5.3	< 1s
	4	9.1	844s	5.7	1364s	5.1	2511s	6.0	< 1s	5.7	< 1s
	6	*	*	*	*	*	*	11.3	< 1s	10.7	< 1s
35	2	5.3	1315s	3	67s	2.7	134s	3.3	< 1s	2.8	< 1s
	4	*	*	*	*	*	*	8.1	< 1s	6.3	< 1s
	6	*	*	*	*	*	*	11.5	< 1s	8.0	< 1s
45	2	*	*	4.1	180s	3	341s	4.6	< 1s	3.8	< 1s
	4	*	*	*	*	*	*	6.7	< 1s	5.1	< 1s
	6	*	*	*	*	*	*	10.7	< 1s	7.7	< 1s

Examining the table, it is quite time-consuming to solve the integer programming models of AMLS, Scheme I, Scheme II. When the number of packets equals or exceeds 6, for networks only with 15 nodes, the integer programming models of Scheme I and Scheme II cannot be solved in one hour, justifying the development of heuristics. The two heuristics clearly demands much less time than solving the integer programming models, and scales well in both network size and the number of packets. Besides, we can also find that the time required for solving integer programming models is much more sensitive to the number of node-pairs than to the number of nodes.

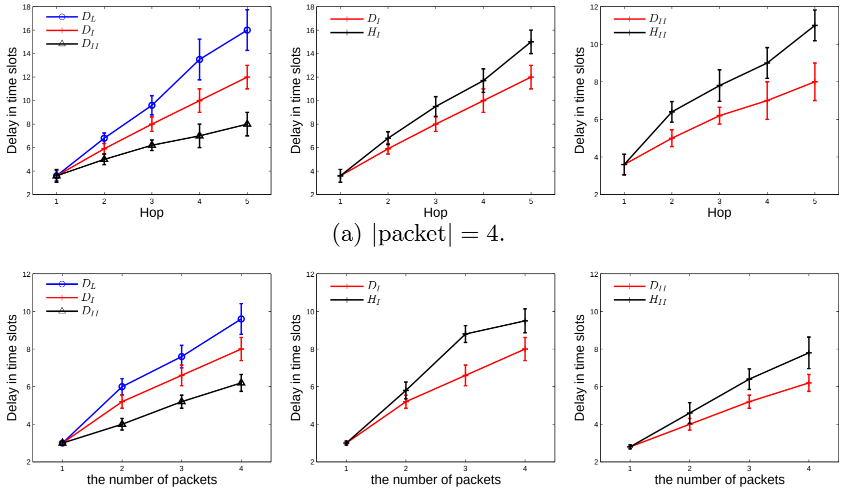
For the delay values over all network instances, we can clearly see that Scheme II is better than Scheme I and both schemes are much better than AMLS. Another observation is that the delay delivered the heuristic is close to that of corresponding exact model.

B. Delay vs hop distance and the number of packets

In the following we compare the performance of the two schemes, as well as show how the delay values relate to the number of packets and the distance (in terms of hops) between the source and the destination of the packets. Here we use small test networks which have 15 nodes. Each delay value is averaged over 10 instances.

Let D_L denote the minimum delay of AMLS, and let H_I and H_{II} denote the delays resulting from the Heuristic I and Heuristic II, i.e., Algorithm 1 and Algorithm 2, respectively. (Recall that D_I and D_{II} denote the minimum delay for S-MDSP and E-MDSP, respectively, see (5a) and (9a).) In Scheme II, when only CF is considered, the minimum delay is denoted by D_{CF} and when only FIC is considered, the minimum delay is denoted by D_{FIC}.

Figure (4a) illustrates the end-to-end delay values of the two schemes versus the source-destination hop distance per packet for 4 packets. The source and destination nodes are randomly selected among node pairs, such that the hop distance equals that under consideration. Let G_{LI}, G_{LII} represent the relative gaps $(D_L - D_I)/D_I \times 100\%$ and $(D_L - D_{II})/D_{II} \times 100\%$ respectively. In Figure (4a), corresponding to hop distance from 2 to 5, G_{LI} are $10.5\%, 15.4\%, 26.7\%, 53.1\%$ and G_{LII} are $31.3\%, 50.0\%, 52.0\%, 88.5\%$. It is evident that Scheme I and Scheme II achieves considerably better end-to-end delay than AMLS, and Scheme II delivers smaller delay than Scheme I. For hop$=5$, the delay delivered by Scheme I is reduced by half compared to AMLS, and Scheme II saves much more than half number of time slots. It verifies the advantages of the two schemes in reducing the delivery delay, and the benefits of CF and FIC.



(a) |packet| = 4.

(b) |hop|=3

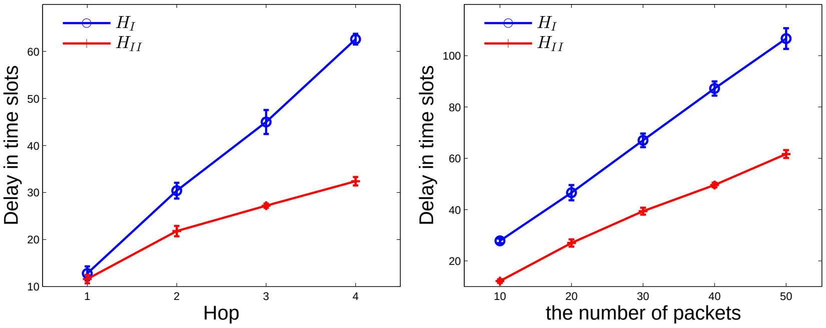
Figure 4: A comparison of the schemes for small networks.

Figure (4b) illustrates the end-to-end delay values of the two schemes versus the number of packets under the condition that the hop distance of each node-pair equals 3. Corresponding to the number of packets $2, 3, 4$, G_{LI} are $15.1\%, 15.4\%, 20\%$ and G_{LII} are $46.1\%, 50.0\%, 54.8\%$. We can clearly see that the relative gaps increases with the number of packets. Again, we observe that Scheme I and Scheme II perform much better than AMLS, and Scheme II is better than Scheme I. Comparing Figure (4a) and Figure (4b), we can also find that Scheme I and Scheme II increases faster with hop distance than with the number of packets.

Figure (4a) and Figure (4b) also present the delay obtained by the two heuristics. We define relative gaps $(H_I - D_I)/D_I \times 100\%$ and $(H_{II} - D_{II})/D_{II} \times 100\%$ for the two heuristics respectively. The maximal gap for the Heuristic I in Figure (4a) is 20.0% while it is 25.0% in Figure (4b). The maximal gap for Heuristic II in Figure (4a) is 26.0% while it is 30.0% in Figure (4b). It shows the heuristics work reasonably well for the two schemes.

Additionally, we present some results for large network instances, which are shown in Figure 5. We consider networks with 50 nodes, which are generated in the same way as before. From Table 3, we have known that the exact optimum of AMLS, Scheme I and Scheme II cannot be obtained in reasonable time for large networks and large number of packets. Therefore, we use the proposed heuristics to compare the performance of the two schemes.

To compare the performance of the two schemes with heuristics, we define the gap of the two heuristics as $G_{III} = (H_I - H_{II})/H_I \times 100\%$. Note that in Figure (4b), G_{III} is 17.9% for |packet|= 4 and |hop|= 3, while G_{III} increases to 42.2% for |packet|= 50 and |hop|= 3 in Figure (5b). G_{III} achieves 48.2% for |packet|= 20 and |hop|= 4 in Figure (5a). This illustrates that Scheme II performs much better than Scheme I for large networks with large number of packets. Again, we can see that the delay values delivered by the two schemes increases with hop distance and the number of packets.



(a) |packet| = 20

(b) |hop| = 3

Figure 5: Comparison of the schemes for large networks.

At last we make an numerical study to show the performance of the two schemes for long-hop node-pairs. We set SINR threshold as $\gamma = 50$. Considering the computation efficiency, we take network instances of 35 nodes and randomly select 3 packets among all node-pairs. The computing results are shown in Figure , in which the maximum hop distance is 7. As we can see, the two schemes perform much better than AMLS for long-hop packets. The gap between the two schemes and AMLS increase with the hop distance, which is consistent with the analysis before.

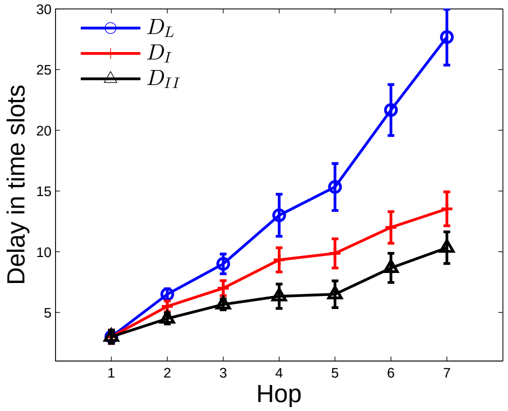


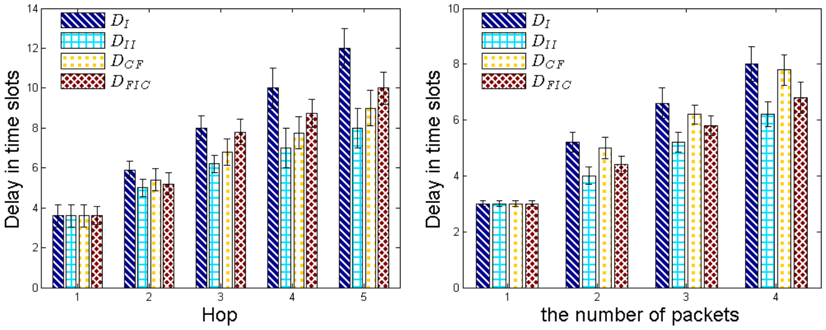
Figure 6: Comparison of the schemes for long-hop instances.

C. The effects of FC and FIC

To show the benefits of CF and FIC separately, we present more results shown in Figure 7. Figure (7a) shows the variation of the end-to-end delay value with the increasing hop distance while Figure (7b) shows the variation of the end-to-end delay value with the increasing number of packets. When CF is used alone, the gap of CF relative to Scheme I, i.e., $(D_I - D_{CF})/D_{CF} \times 100\%$ ranges from 0% to 14.3% in (7a) and ranges from 0% to 6.5% in (7b). Similarly, for FIC, the gap, i.e., $(D_I - D_{FIC})/D_{FIC} \times 100\%$ ranges from 0% to 33.3% in (7a) and ranges from 0% to 18.1% in (7b). Further,the gap of CF relative to Scheme II ranges from 0% to 20.1% in (7a) and ranges from 0% to 20.5% in (7b). The gap of FIC relative Scheme II ranges from 0% to 11.3% in (7a) and ranges from 0% to 10.3% in (7b). We can see that only CF or only FIC can help to reduce the delay compared with Scheme I. Combining CF and FIC together, which is Scheme II, the delay is significantly reduced. In Figure (7a), we observe that CF performs better than FIC at $|\text{hop}|= 2$ while not at other hop distances. Thus we cannot say that CF is better than FIC or not.

E. Transmission parallelization

We have compared the two schemes with AMLS from the viewpoint of the



(a) |packet| = 4

(b) |hop| = 3

Figure 7: The effects of CF and FIC.

delay value and concluded that the two schemes greatly outperform AMLS. Here we compare the two schemes in terms of transmission parallelization, measured as the average number of transmissions in a compatible set, i.e., the average number of links over the compatible sets used in a scheduling solution. For example, the average number of transmissions per compatible set scheduling solutions in Figure 3 are $(2+2+2+2+2)/5 = 2$, $(2+2+1+1+1+1)/6 = 1.3$, $(2+3+3+1+1)/5 = 2$ and $(2+2+1+2+2)/5 = 1.8$ respectively.

The results for networks of various sizes are shown in Fig. 8. For this figure, the number of packets equals 3, which are selected randomly among all node-pairs.Each value in the figure is the average of 10 network instances.

We observe that AMLS has a larger average number of transmissions per compatible set than Scheme I, and increases in network size. For Scheme I, the number of parallel transmissions per time slot is fairly constant with respect to network size. Thus optimizing delay is very different from maximizing spatial reuse of transmissions, demonstrating the significance of schemes tailored for delay-driven scheduling. However, Scheme II is better than AMLS in terms of transmission parallelization. The reason is that CF and FIC in Scheme II enables more transmissions per compatible set.

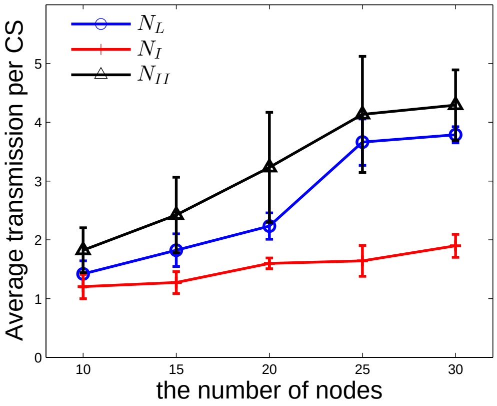


Figure 8: Average number of transmissions per compatible set.

8. Conclusions

We have investigated the problem of minimizing end-to-end delay in wireless networks under the physical interference model from an optimization viewpoint. We have proposed two schemes. Scheme I is a novel approach for delay-minimization scheduling, providing an efficient instrument to fully optimize the delay. Scheme II incorporates two advanced physical-layer techniques, i.e., cooperative forwarding and forward interference cancellation. We have presented a set of novel mathematical formulations for the two schemes and provided insight into computational complexity and solution characterization, as well as faster heuristics. The numerical results illustrate the solutions of the two schemes, and show the benefits brought by CF and FIC.

Acknowledgments

The research reported in this paper was funded by European Union FP7 Marie Curie projects 324515 and 329313. M. Pióro was supported by the National Science Centre, Poland, under the grant no. 2017/25/B/ST7/02313, “Packet routing and transmission scheduling optimization in multi-hop wireless networks with multicast traffic”.

Appendix A. The proof of Theorem 2.

Proof. We prove the result by a reduction from the classical $\mathcal{NP}$-complete 3-satisfiability problem (3-SAT) [35]. A 3-SAT instance consists of B Boolean variables $\{v_1, v_2, \ldots, v_B\}$ and C clauses $\{c_1, c_2, \ldots, c_C\}$. For each variable v_i, $i = 1, 2, \ldots, B$, we use $\overline{v}_i$ to denote its negation. The variables and their negations are referred to as literals. Each clause c_l is a conjunction of three literals, for example $v_i \vee v_j \vee \overline{v}_k$. The decision problem is to determine whether or not there exists an assignment $Q : \{v_1, v_2, \ldots, v_B\} \rightarrow \{true, false\}$ (assigning the *true* or *false* value to each variable) such that all the clauses hold true.

For the reduction, we construct an instance of a decision version of S-MDSP that corresponds to a given instance of 3-SAT. The network graph underlying the instance has $2B + 6C$ links. For every variable v_i and its negation $\overline{v}_i$, we define "literal" links (v_i, v_i'), and $(\overline{v}_i, \overline{v}_i')$, respectively. For every clause $c_l, l = 1, 2, \ldots, C$, we introduce, for each of the three literals of c_l, two "clause" links in series, such that the two links form a two-hop path. The three paths defined for the three literals are disjoint, and start and end at the same nodes c_l and c_l'', respectively. Suppose clause c_l contains literal v_i. Then the two corresponding links are $(c_l, c'_{l,i})$ and $(c'_{l,i}, c_l'')$, respectively. For a literal that is a negation, e.g., $\overline{v}_i$, we use $(c_l, c'_{l,\bar{i}})$ and $(c'_{l,\bar{i}}, c_l'')$ to denote the two links in question. As there are three literals per clause, there are $6C$ clause links in total.

Next we introduce $2B$ compatible sets, all containing one literal link and a number of clause links (as specified below). Each of the $2B$ literal links appears in exactly one compatible set. The compatible set containing link (v_i, v_i') will be denoted by v_i, and the compatible set containing link $(\overline{v}_i, \overline{v}_i')$ by $\overline{v}_i$ (identifying the compatible sets by the literals will not lead to confusion). The clause links are included into the compatible sets in the following way. Consider an arbitrary clause c_l and suppose v_i is one of the its literals. Then, link $(c_l, c'_{l,i})$ is added to compatible set v_i, and link $(c'_{l,i}, c_l'')$ is added to compatible set $\overline{v}_i$. If the literal is a negation of a variable, say $\overline{v}_k$, then $(c_l, c'_{l,\overline{k}})$ is added to compatible set $\overline{v}_k$, and $(c'_{l,\overline{k}}, c_l'')$ is added to compatible set v_k. In effect, for every clause, each of its six clause links is added to one of six different compatible sets. (Note that we assume that a variable and its negation do not appear together in any clause, as such a clause is always true and thus can be eliminated from the 3-SAT instance.) An illustration of the reduction is given in Figure A.9

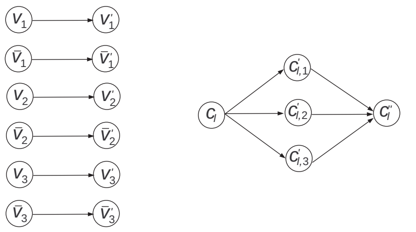


Figure A.9: An illustration of the literal links defined for three variables v_1, v_2, v_3 and their negations $\overline{v}_1, \overline{v}_2, \overline{v}_3$. The example clause c_l is composed by $(v_1 \vee \overline{v}_2 \vee v_3)$. One compatible set is defined with respect to each literal link. For the example clause, link $(c_l, c_{l,1}')$ is compatible with (v_1, v_1'), and $(c_{l,1}', c_l'')$ is compaible with $(\overline{v}_1, \overline{v}_1')$. Similar construction applies for the compatible sets defined for variable v_3, $\overline{v}_3$, and clause c_l. For v_2 and $\overline{v}_2$, link $(c_l, c_{l,2}')$ is compatible with $(\overline{v}_2, \overline{v}_2')$, and $(c_{l,1}', c_l'')$ is compaible with (v_2, v_2').

Finally, we introduce a set $\mathcal{S}$ of $2B + C$ packets, referred to as "literal" and "clause" packets, respectively. The source and destination of a literal packet v_i $(i = 1, 2, \ldots, B)$ are v_i and v_i', respectively. Thus, routing of a literal packet is fixed and involves a single-hop transmission on its literal links. A similar construction applies to literal packets $\overline{v}_i, i = 1, 2, \ldots, B$. For a clause packet corresponding to $c_l, l = 1, 2, \ldots, C$, the source is c_l, and the destination is c_l''. Thus a clause packet can take any of the three paths, each having two clause links as defined above.

For the introduced network setting, the minimal number of time slots required for Scheme I to deliver all packets to their destinations (i.e., D_I^* – the optimal objective value of S-MDSP) is equal to or greater than $2B$, since to deliver all of the $2B$ literal packets we have to apply all $2B$ compatible sets, each used in exactly one time slot. The question is whether using each of the $2B$

compatible exactly once is also sufficient to deliver all of the C clause packets. This leads to the decision version of S-MDSP (referred to as S-MDSP-D): *for the defined network setting, does $D_I^* = 2B$ hold?*

Consider an arbitrary a permutation of the $2B$ compatible sets where $a(w)$ denotes the index of the time slot that applies compatible set w (where w is of the form v_i or $\overline{v}_i$ for some $i = 1, 2, \ldots, B$). Clearly, the literal packet w will be delivered in the time slot $t = a(w)$, so the literal packets experience delays of $2B$ slots or less. A clause packet, however, can experience delay that is greater than $2B$ for all its three possible route choices. If we consider a clause c_l of the form $v_i \vee v_j \vee \overline{v}_k$, then the delay of its packet will be greater than $2B$ if, and only if, $a(\overline{v}_i) < a(v_i)$, $a(\overline{v}_j) < a(v_j)$ and $a(v_k) < a(\overline{v}_k)$. This is because for all the three routes of packet c_l, the second link is scheduled before the first one so one frame of length $2B$ is not sufficient to deliver the packet (note that the packet will be delivered in the time slot equal to $2B + \min\{a(\overline{v}_i), a(\overline{v}_j), a(v_k)\}$). Otherwise, the packet will be delivered during one frame of $2B$ slots (the exact number of the time slot during which the packet is delivered is straightforward to determine also in this case). Hence, from the viewpoint of the S-MDSP-D question, it is sufficient to consider only the permutations a with compatible sets v_1 and $\overline{v}_1$ occupying the first two slots (in any order), compatible sets v_2 and $\overline{v}_2$ occupying the next two slots (in any order), and so on. Note that the number of such "normalized" permutations is 2^B, and so is the number of possible assignments for 3SAT.

Finally, we define the one-to-one correspondence of assignments Q and the normalized permutations a: $Q \leftrightarrow a$ if, and only if, for each v_i with $Q(v_i) = true$, $a(v_i) = 2i - 1, a(\overline{v}_i) = 2i$, and for each v_j with $Q(v_j) = false$, $a(v_j) = 2j, a(\overline{v}_i) = 2j - 1$. The crucial property of this correspondence is that a clause c_l $(i = 1, 2, \ldots, C)$ is satisfied by Q if, and only if, the packet corresponding to c_l is delivered during the first frame. Thus, 3-SAT reduces to S-MDSP-D. □

References

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

[2] O. Goussevskaia, R. Wattenhofer, M. M. Halldorsson, E. Welzl, Capacity of Arbitrary Wireless Networks, in: IEEE INFOCOM, 2009, pp. 1872–1880.

[3] M. Dinitz, Distributed Algorithms for Approximating Wireless Network Capacity, in: IEEE INFOCOM, 2010, pp. 1–9.

[4] P. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, S. Tang, Wireless Link Scheduling under Physical Interference Model, in: IEEE INFOCOM, 2011, pp. 838–845.

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

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

[7] A. Capone, G. Carello, Scheduling Optimization in Wireless MESH Networks with Power Control and Rate Adaptation, in: IEEE SECON 2006, 2006.

[8] P. Björklund, P. Värbrand, D. Yuan, Resource Optimization of Spatial TDMA in Ad hoc Radio Networks: a Column Generation Approach, in: IEEE INFOCOM, 2003, pp. 818–824.

[9] M. Doudou, D. Djenouri, N. Badache, Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks, IEEE Communications Surveys Tutorials 15 (2) (2013) 528–550.

[10] A. Capone, S. Gualandi, L. Chen, D. Yuan, A New Computational Approach for Maximum Link Activation in Wireless Networks under the S-INR model, IEEE Transactions on Wireless Communicaitons 10 (5) (2011) 1368–1372.

[11] V. Angelakis, A. Ephremides, Q. He, D. Yuan, Minimum-Time Link Scheduling for Emptying Wireless Systems: Solution Characterization and Algorithmic Framework, IEEE Transactions on Information Theory 60 (2) (2014) 1083–1100.

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

[13] M. Pióro, M. Żotkiewicz, B. Staehle, D. Staehle, D. Yuan, On Max-min Fair Flow Optimization in Wireless Mesh Networks, Ad Hoc Networks 13 (0) (2014) 134–152.

[14] M. Żotkiewicz, Max-Min Fairness in WMNs with Interference Cancelation Using Overheard Transmissions, Journal of Applied Mathematics 2014.

[15] L. Bui, R. Srikant, A. Stolyar, Novel Architectures and Algorithms for Delay Reduction in Back-pressure Scheduling and Routing, in: IEEE INFOCOM, 2009, pp. 2936–2940.

[16] G. R. Gupta, N. B. Shroff, Delay Analysis and Optimality of Scheduling Policies for Multihop Wireless Networks, IEEE/ACM Transactions on Networking 19 (1) (2011) 129–141.

[17] B. Ji, C. Joo, N. B. Shroff, Delay-based Back-pressure Scheduling in Multihop Wireless Networks, IEEE/ACM Transactions on Networking 21 (5) (2013) 1539–1552.

[18] A. Saifullah, Y. Xu, C. Lu, Y. Chen, End-to-End Communication Delay Analysis in Industrial Wireless Networks, IEEE Transactions on Computers 64 (5) (2015) 1361–1374.

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

[20] D. Gong, Y. Yang, Low-latency SINR-based Data Gathering in Wireless Wensor Networks, in: IEEE INFOCOM, 2013, pp. 1941–1949.

[21] R. Gandhi, Y. Kim, S. Lee, J. Ryu, P. Wan, Approximation Algorithms for Data Broadcast in Wireless Networks, IEEE Transactions on Mobile Computing 11 (7) (2012) 1237–1248.

[22] P. Djukic, S. Valaee, Link Scheduling for Minimum Delay in Spatial Re-Use TDMA, in: IEEE INFOCOM, 2007, pp. 28–36.

[23] A. Capone, S. Gualandi, 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) (2011) 1256–1269.

[24] A. Sendonaris, E. Erkip, B. Aazhang, User Cooperation Diversity, Part I: System Description/Part II: Implementation Aspects and Performance Analysis, IEEE Transactions on Communications 51 (11) (2003) 1927–1948.

[25] X. Tao, X. Xu, Q. Cui, An Overview of Cooperative Communications, IEEE Communications Magazine 50 (6) (2012) 65–71.

[26] S. Zhang, S. Liew, H. Wang, Blind Known Interference Cancellationn, IEEE Journal on Selected Areas in Communications 31 (8) (2013) 1572–1582.

[27] C. Qin, N. Santhapuri, S. Sen, S. Nelakuditi, Known Interference Cancellation: Resolving Collisions Due to Repeated Transmissions, in: Fifth IEEE Workshop on Wireless Mesh Networks (WIMESH), 2010, pp. 1–6.

[28] N. Michelusi, P. Popovski, O. Simeone, M. Levorato, M. Zorzi, Cognitive Access Policies under a Primary ARQ Process via Forward-Backward Interference Cancellation, IEEE Journal on Selected Areas in Communications 31 (11) (2013) 2374–2386.

[29] E. C. van der Meulen, Three-terminal Communication Channels, Advanced Appliled Probability 3 (1971) 120–154.

[30] T. M. Cover, A. A. E. Gamal, Capacity Theorems for the Relay Channel, IEEE Transactions on Information Theory 25 (5) (1979) 572–584.

[31] D. Yuan, V. Angelakis, L. Chen, E. Karipidis, E. G. Larsson, On optimal Link Activation with Interference Cancellation in Wireless Networking, IEEE Transactions on Vehicular Technology 62 (2) (2013) 939–945.

[32] D. Halperin, T. Anderson, D. Wetherall, Taking the Sting out of Carrier Sense: Interference Cancellation for Wireless LANs, in: ACM MOBICOM, 2008, pp. 339–350.

[33] S. Gollakota, D. Katabi, Zigzag Decoding: Combating Hidden Terminals in Wireless Networks, in: ACM SIGCOMM, 2008, pp. 159–170.

[34] X. Zhang, K. G. Shin, Chorus: Collision Resolution for Efficient Wireless Broadcast, in: IEEE INFOCOM, Piscataway, NJ, USA, 2010, pp. 1747–1755.

[35] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.