

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 Department of Electronics Information and Bioengineering, Technical University of Milan, 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

ARTICLE INFO

Article history:

Received 28 April 2018

Revised 26 November 2018

Accepted 24 January 2019

Available online 8 February 2019

Keywords:

Cooperative forwarding

Forward interference cancellation

Link scheduling

Mathematical programming

SINR Model

Wireless networks

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.

© 2019 Elsevier B.V. All rights reserved.

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 applications have these characteristics: sensor networks with large scale deployments, data gathering and processing for industrial ap-

plications, 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.

* Corresponding author.

E-mail addresses: capone@elet.polimi.it (A. Capone), yuan.li@nudt.edu.cn (Y. Li), mpp@tele.pw.edu.pl (M. Pióro), di.yuan@liu.se (D. Yuan).

In the literature, the problem of scheduling in single hop and multi-hop wireless networks has been studied extensively, see, e.g., [1–3]. 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 in the assumed periodic time frame [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] so that a particular 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. Yet, 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, that is, 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 cancellation. 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. In Section 2 we review related work. Section 3 is devoted to the network model and the considered delay-minimization schemes; it also contains an example illustrating the main idea behind the schemes. Optimization problem formulations, complexity analysis, and solution approaches for Schemes I and II are presented in Sections 4 and 5, respectively. Then, in Section 6, numerical results are discussed, followed by concluding remarks in Section 7.

2. 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 [15–17], 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 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 studied the problem of maximum link activation with CF and interference cancellation. The problem setup is related to ours, and, to some extent, it resembles the optimization decisions in a single time slot in our problem. However, there are significant differences. First, the problem in [23] has no notion of time or any aspect of routing, and hence the approaches in [23] are not able to address end-to-end delay, and in fact, the study in [23] is for a single-hop system. Second, interference cancellation in [23] is idealized in the sense that a receiver can cancel any interference if the interference is strong enough. This is different from FIC for which nodes have received which packets are of significance in interference cancellation. In [24], 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 [25,26]. 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 [27], 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 [28]. The authors of [29] show the benefit of FIC for a two-user scenario in cognitive radio networks.

Finally, we would like to clarify here that our work does not consider scenarios with dynamic traffic where the problem is mainly that of regulating channel access of packet transmissions (see [30] for a survey). In the network scenario in our work, we assume that channel and network (traffic) conditions are predictable and stable. The variability of the channel is low so that channel estimation can be performed not too often and its cost can be neglected. For these network scenarios, a central controller is able to perform scheduling/routing optimization and then distribute it to all nodes. Since traffic and channel conditions are almost stable, the related control overhead can be assumed to be negligible. The use of a central control entity is considered for example in wireless

networks based on the software defined network paradigm (see e.g. [31]).

3. Network model

3.1. 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 $\mathcal{T} := \{1, 2, \dots, T\}$. 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:

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

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, when CF or FIC is not applied, 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. In the following we will refer the so described transmission model as *standard packet forwarding*.

A set $c \subseteq \mathcal{A}$ of arcs that can be simultaneously active with respect to the above standard packet forwarding 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 $(i, j) \in c$:

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

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 $t = \bar{T}$ 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.

3.2. Cooperative Forwarding (CF)

The idea of *cooperative packet forwarding* has been studied in recent years. CF dates back to the work of [32,33]. 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 [24], in which the SINR condition for node $j \in \mathcal{V}$ to correctly receives packet $s \in \mathcal{S}$ is as follows:

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

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 ($\mathcal{T} := \{1, 2, \dots, T\}$) |

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 .

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 3.1. 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.

3.3. Forward interference cancellation (FIC)

In multi-hop wireless networks, it is common that a node receives or overhears a packet several times. The *forward interference cancellation* technique, aimed at addressing interference due to repeated transmissions of the same packet, has been proposed in [28]. 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 [28], FIC is different from the widely studied successive interference cancellation (SIC) scheme [34]. 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 [28] 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) [35], ZigZag [36] and Chorus [37], 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:

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

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

3.4. 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 next two sections we will study two versions of MDSP, with standard packet forwarding (S-MDSP), and the extension with CF/FIC packet forwarding (E-MDSP), using exact integer programming models as well as heuristic methods.

3.5. An illustrative example

It is instructive to compare, using an example, the delay incurred by the conventional minimum-length schedule with repeated frames, versus that of the two new schemes for packet delay minimization. Such an example is given in Fig. 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 under standard packet forwarding. There are two packets, white and black; the white packet is to be delivered from node a to node e, while the black one from e to a. By assumption, it takes exactly one time slot to transmit a packet over a link. Clearly, under standard packet forwarding each of the two considered packets needs four link transmissions to be delivered from its source to its destination.

It is easy to notice that with conventional minimum-length scheduling, the minimum frame length is four time slots, with the corresponding compatible sets $\{1, 3\}$, $\{2, 4\}$, $\{6, 8\}$, $\{5, 7\}$, each including two links that are exactly two hops away. In order to deliver the two packets, the frame must be repeated two times. Thus, 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 the second slot of the first frame, and on links 3 and 4 in the first and the 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 the fourth slot of the first frame, and on links 6 and 5 in the third and the fourth slot of the second frame).

Observe that in the considered setting there are some idle link transmissions. In the first frame these are the transmissions on link 3 in the first slot, on link 4 in the second slot, on link 6 in the third slot, and on link 5 in the fourth slot. In the second frame the transmissions on links 1, 2, 7, 8 in their respective slots are

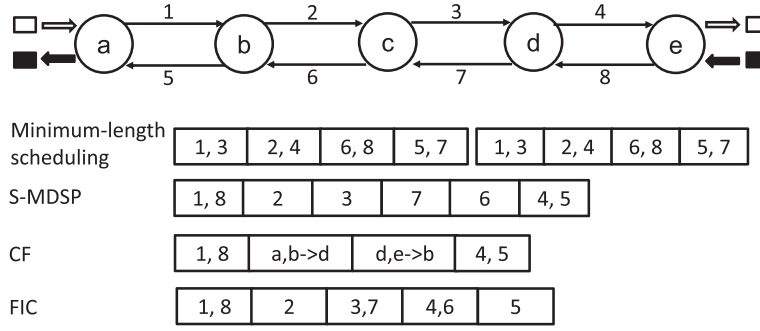


Fig. 1. An illustration of the end-to-end delay.

idle. However, if we assume that successive instances of the white and the black packet appear in the network at the beginning of each successive time frame, then each such instance will follow its predecessor, and all frames except the first one will be fully occupied. In other words, the packets of each color will form “a train” of packets, with each successive packet delivered to its destination during six slots (white slots) or eight slots (black packets). This phenomenon is explained in detail in [38].

When it comes to minimization of the maximum packet delay in a single frame, for S-MDSP the optimum is a frame of length six. As indicated in Fig. 1, in the optimal frame the first and the last compatible set includes two links, {1, 8} and {4, 5}, while the other compatible sets, {2}, {3}, {7}, {6}, include only one link each. Obviously, the frame is longer than the previous one (when the frame length is minimized), but allows to deliver both packets in six slots. This gain, however, is achieved at an expense of reducing the carried traffic when the periodic packet streams are considered. In the conventional minimum-length scheduling it was 2 packets per 4 time slots (i.e., per frame), and now it is 2 packets per 6 time slots.

For E-MDSP, the effects of CF and FIC are shown separately. With CF, the delay for both white and black packets is reduced by two in comparison with the optimal delay of S-MDSP: 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 the 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 the 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 S-MDSP. Observe that E-MDSP with FIC carries 2 packets per 5 slots, while E-MDSP with CF carries 2 packets per 4 slots (i.e., maximal carried traffic achievable with standard packet forwarding) with the packet delay equal to 4 slots (i.e., minimal delay achievable with standard packet forwarding for 4-hop packets).

4. Scheme I: Minimum delay scheduling with standard packet forwarding

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

4.1. Formulation of S-MDSP

Let T denote an upper bound on the delay, with $\mathcal{T} = \{1, 2, \dots, T\}$. 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_{is}^t – equal to 1 if node $i \in \mathcal{V}$ is sending $s \in \mathcal{S}$ in $t \in \mathcal{T}$; Y_{is}^t – equal to 1 if $i \in \mathcal{V}$ is receiving $s \in \mathcal{S}$ in $t \in \mathcal{T}$; y_{is}^t – 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.

$$\text{minimize } D_I = \sum_{t \in \mathcal{T}} \lambda^t \quad (5a)$$

$$\lambda^{t+1} \leq \lambda^t, \quad t \in \mathcal{T} \setminus \{T\} \quad (5b)$$

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

$$\sum_{i \in \mathcal{V}} X_{is}^t \leq 1, \quad \sum_{i \in \mathcal{V}} Y_{is}^t \leq 1, \quad s \in \mathcal{S}, t \in \mathcal{T} \quad (5d)$$

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

$$\sum_{t \in \mathcal{T}} X_{is}^t \leq 1, \quad \sum_{t \in \mathcal{T}} Y_{is}^t \leq 1, \quad i \in \mathcal{V}, s \in \mathcal{S} \quad (5f)$$

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

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

$$X_{is}^t \leq y_{is}^t, \quad i \in \mathcal{V}, s \in \mathcal{S}, t \in \mathcal{T} \quad (5i)$$

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

$$\lambda, X, y, Y \text{ binary.} \quad (5k)$$

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

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

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.

4.2. 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]. \square

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.*

4.3. A heuristic algorithm for S-MDSP

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.

Algorithm 1 A heuristic for S-MDSP.

Step 0: Put $\mathcal{S}(1) := \mathcal{S}$ (as nopacket 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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

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). Eq. (8b) records the packets that have not been delivered to their destinations in time slot $t+1$. Eq. (8c)

Table 2
Notations for the heuristic.

| Notation | Description |
|----------------------|---|
| $S(t)$ | Set of packets not yet delivered to their respective destinations by the beginning of slot t |
| $S(i, t)$ | Set of packets from $S(t)$ present at node $i \in \mathcal{V}$ |
| $S'(i, t)$ | $:= S(t) \setminus S(i, t)$, set of packets from $S(t)$ 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 $s \in S(t)$ 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 $s \in S(t)$, and 0 otherwise |
| Y_{js} | Binary variable that is equal to 1 if node $j \in \mathcal{V}'(s, t)$ is receiving packet $s \in S(t)$, 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 $s \in S(t)$ after the transmission |
| Z_{0s} | Auxiliary binary variable for $s \in S(t)$ |
| z_s | Variable expressing the distance to destination of packet $s \in S(t)$ after the transmissions in slot t |

expresses the set of packets contained in each node. Eq. (8d) computes for each packet the minimal distance to its destination.

5. 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.

5.1. 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).

$$\text{minimize } D_{II} = \sum_{t \in \mathcal{T}} \lambda_t \quad (9a)$$

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

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

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

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

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

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

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

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

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

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

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

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 (9e) is skipped as it is implicitly included in the new SINR constraint (9j). Constraint (9f) 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 $\gamma(\eta + \sum_{k \in \mathcal{V} \setminus \{j\}} p(k, j))$ – 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.

5.2. 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. \square

5.3. 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 4.3. 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).

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

Table 3
A comparison of delay and computing time.

| V | S | AMLS | | S-MDSP | | E-MDSP | | S-Heuristic | | E-Heuristic | |
|----|---|-------|---------|--------|---------|--------|---------|-------------|---------|-------------|---------|
| | | 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 |

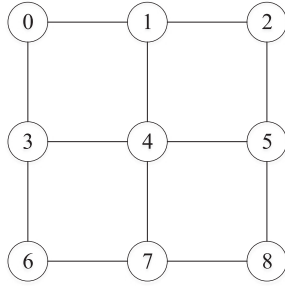


Fig. 2. A grid network example.

(7b), (7d), (7e), (7f)

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

X, Y, Z binary, z continuous. (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.

6. 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 $P = .1$ 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 [39], referred to as solution driven by minimum-frame-length scheduling, which minimizes the end-to-end 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).

6.1. An illustrative example

We use a grid network depicted in Fig. 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 $\gamma = 10$.

Fig. 3 shows scheduling solutions of AMLS, S-MDSP and E-MDSP. Each square corresponds to a time slot and indicates the transmitting links. Packet $s = 1$ is transmitted along links marked with underline while packet $s = 2$ is transmitted along bold italic

| frame | | | | | frame | | | | |
|----------------|----------------|------|------|----------------|----------------|----------------|----------------|----------------|------|
| <u>2->1</u> | 0->3 | 3->6 | 3->0 | <u>1->0</u> | 2->1 | <u>0->3</u> | <u>3->6</u> | <u>3->0</u> | 1->0 |
| 6->3 | <u>8->7</u> | 5->8 | 5->2 | <u>7->6</u> | <u>6->3</u> | 8->7 | 5->8 | 5->2 | 7->6 |

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

| | | | | | |
|----------------|----------------|------|------|----------------|----------------|
| <u>2->1</u> | <u>1->0</u> | 6->3 | 3->0 | <u>0->3</u> | <u>3->6</u> |
| 8->7 | 7->6 | | | | |

(b) S-MDSP: delay = 6 time slots.

| | | | | |
|----------------|------------|---------|---------|----------------|
| <u>2->1</u> | <u>1,2</u> | 7,8 | 4,5,6,7 | <u>0,1,4,5</u> |
| 8->7 | ->0,4,5 | ->4,5,6 | ->0 | ->6 |

(c) CF: delay = 5 time slots.

| | | | | |
|------------------|----------------|------|----------------|----------------|
| <u>2->1,5</u> | <u>1->0</u> | 5->2 | <u>0->3</u> | <u>3->6</u> |
| 8->5 | 2->1 | 1->0 | | |

(d) FIC: delay = 5 time slots.

Fig. 3. The frames optimized for different assumptions.

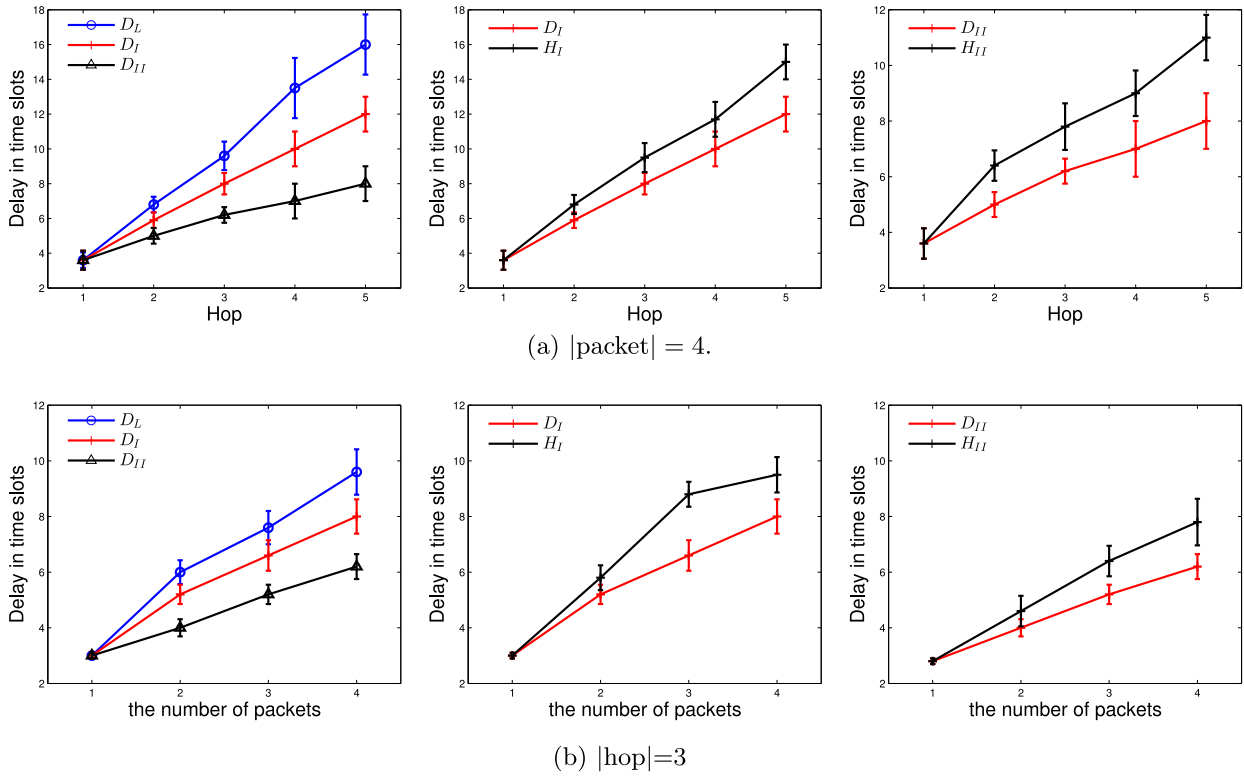


Fig. 4. A comparison of the schemes for small networks.

links. With AMLS, we find a frame composed of a list of ordered compatible sets, shown as the frame in Fig. 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.

AMLS mainly improves transmission parallelization, i.e., parallelizing as many transmissions as possible, which is not optimal from the delay standpoint. For S-MDSP, 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 Fig. 3b, the delay is 6 time slots, which saves 3 time slots compared to Fig. 3a. With CF, the delay is further reduced by 1, see Fig. 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 S-MDSP. In time slot 4 of Fig. 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 $s = 1$.

6.2. 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 $1000 \text{ m} \times 1000 \text{ m}$. The SINR threshold is set as $\gamma = 10$.

A. Computational efficiency

We present the computational efficiency for solving the optimization models of AMLS, S-MDSP, E-MDSP and the two heuristics (S-Heuristic and E-Heuristic) 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.

Examining the table, it is quite time-consuming to solve the integer programming models of AMLS, S-MDSP, E-MDSP. When the number of packets equals or exceeds 6, for networks only with 15 nodes, the integer programming models of S-MDSP and E-MDSP cannot be solved in one hour, justifying the development of heuristics. The two heuristics clearly demand much less time than solving the integer programming models, and scale 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 packets than to the number of nodes.

For the delay values over all network instances, we can clearly see that E-MDSP is better than S-MDSP and both schemes are much better than AMLS. Another observation is that the delay values delivered the heuristics are close to that of corresponding exact models.

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. Let D_I and D_{II} denote the minimum delay for S-MDSP and E-MDSP, respectively,

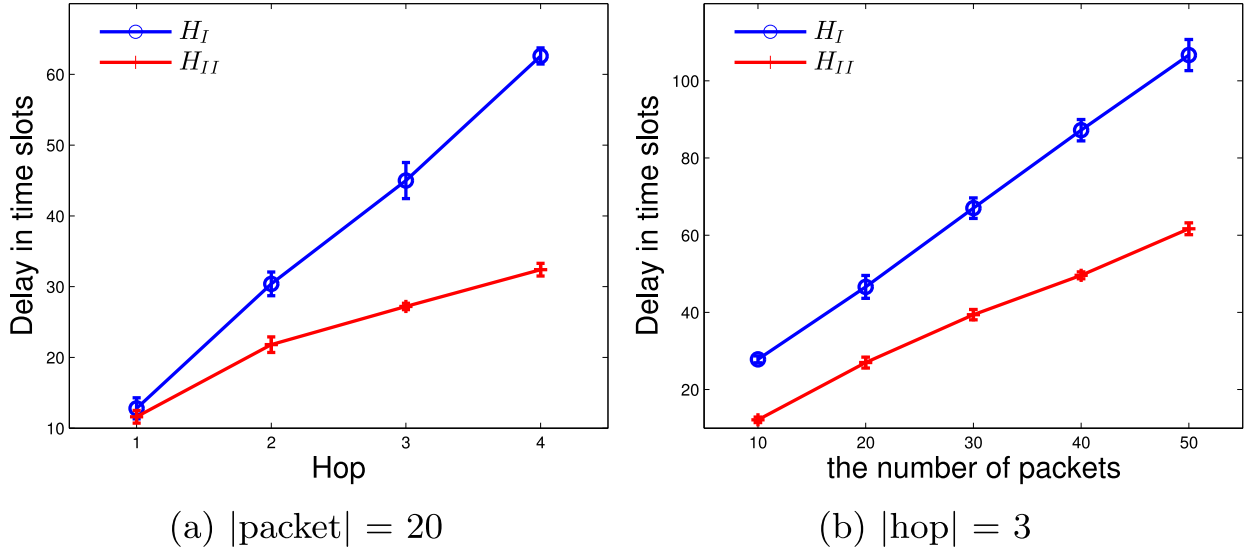


Fig. 5. Comparison of the schemes for large networks.

see (5a) and (9a). Let H_I and H_{II} denote the delays resulting from S-Heuristic and E-Heuristic, respectively. In E-MDSP, 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} .

Fig. 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 $(D_L - D_I)/D_L \times 100\%$ and $(D_L - D_{II})/D_L \times 100\%$ respectively. In Fig. 4a, corresponding to hop distance from 2 to 5, G_{LI} are 2.9%, 12.5%, 21.1%, 25.7% and G_{LII} are 26.5%, 35.4%, 48.1%, 50.9%. It is evident that S-MDSP and E-MDSP achieves considerably better end-to-end delay than AMLS, and E-MDSP delivers smaller delay than S-MDSP. For $\text{hop}=5$, the delay delivered by S-MDSP is reduced by one fourth compared to AMLS, and E-MDSP saves around 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.

Fig. 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. For 2, 3 and 4 packets respectively, G_{LI} are 13.3%, 15.4%, and 16.7%, and G_{LII} are 33.3%, 34.5%, and 35.8%. We can clearly see that the relative gaps increase with the number of packets. Again, we observe that S-MDSP and E-MDSP perform much better than AMLS, and E-MDSP is better than S-MDSP. Comparing Fig. 4 and b, we can also find that the delays delivered by S-MDSP and E-MDSP increase faster with hop distance than with the number of packets.

Fig. 4a and b 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 S-Heuristic in Fig. 4a is 20.0% while it is 25.0% in Fig. 4b. The maximal gap for E-Heuristic in Fig. 4a is 26.0% while it is 30.0% in Fig. 4b. It shows that the heuristics work reasonably well for the two schemes.

Additionally, we present some results for larger network instances, which are shown in Fig. 5. We consider networks with 50 nodes, which are generated in the same way as before. From Table 3, we know that the exact optimum of AMLS, S-MDSP, and E-MDSP 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. We define the gap of the two heuristics as $G_{III} = (H_I - H_{II})/H_I \times 100\%$.

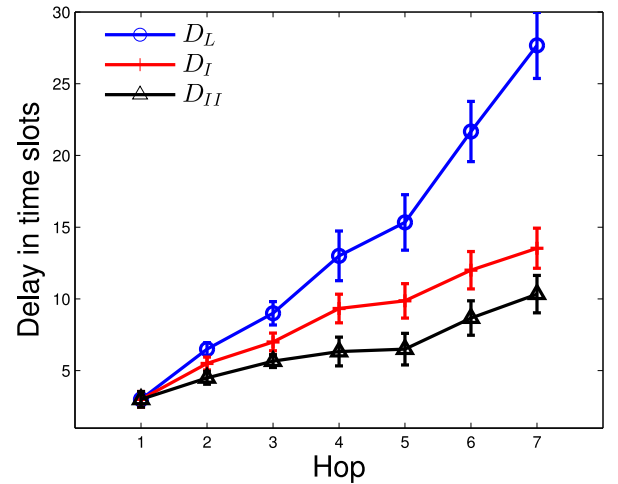


Fig. 6. Comparison of the schemes for long-hop instances.

Note that in Fig. 4b, G_{III} is 17.9% for $|\text{packet}|=4$ and $|\text{hop}|=3$, while G_{III} increases to 42.2% for $|\text{packet}|=50$ and $|\text{hop}|=3$ in Fig. 5b. G_{III} achieves 48.2% for $|\text{packet}|=20$ and $|\text{hop}|=4$ in Fig. 5a. This illustrates that E-MDSP performs much better than S-MDSP for large networks with large number of packets. Again, we can see that the delay values delivered by the two schemes increase with hop distance and the number of packets.

At last we make a 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 Fig. 6, 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 increases with the hop distance, which is consistent with the analysis before.

C. The effects of FC and FIC

To show the benefits of CF and FIC separately, we present more results in Fig. 7. Fig. 7a shows the variation of the delay values with the increasing hop distance while Fig. (7)b shows the variation of the delay values with the increasing number of packets. When CF is used alone, the gap of CF relative to S-MDSP, i.e.,

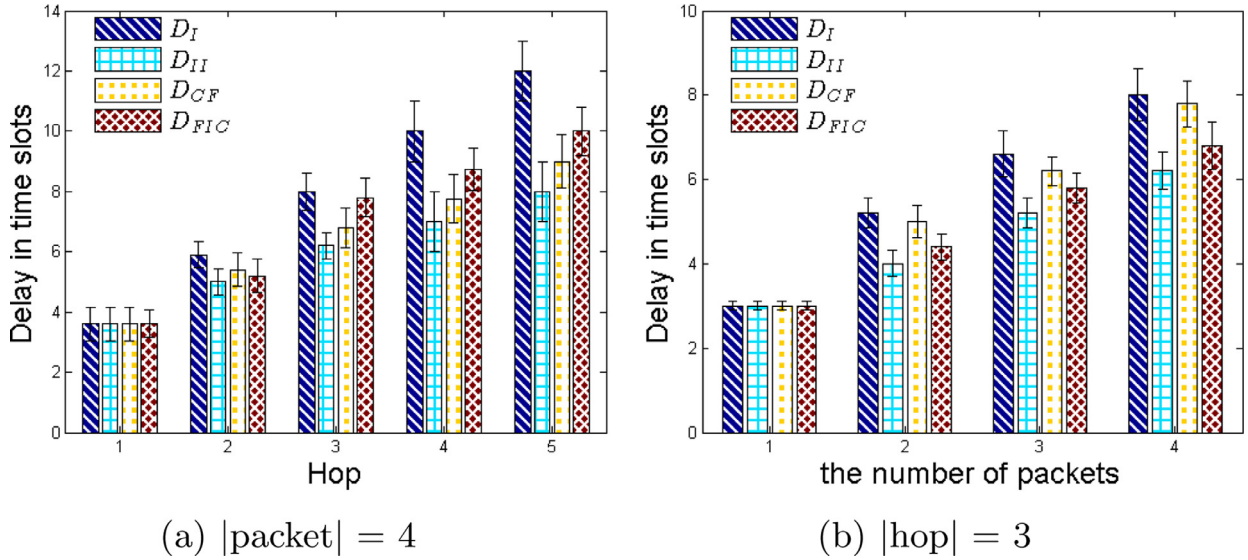


Fig. 7. The effects of CF and FIC.

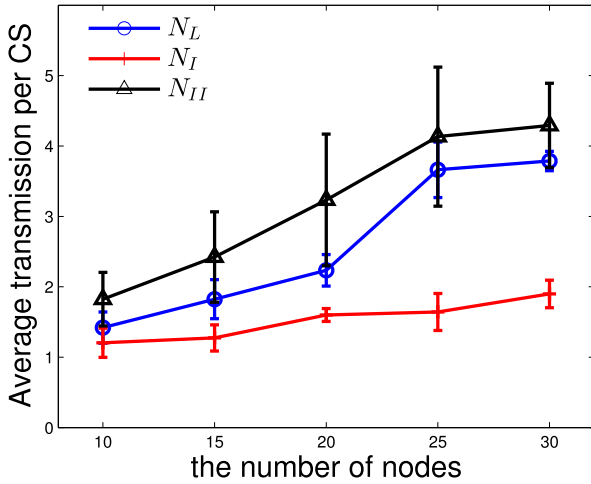


Fig. 8. Average number of transmissions per compatible set.

$(D_I - D_{CF})/D_I \times 100\%$ ranges from 0% to 14.3% in (7)a and ranges from 0% to 6.5% in (7)b. Similarly, for FIC, the gap, i.e., $(D_I - D_{FIC})/D_I \times 100\%$ ranges from 0% to 33.3% in (7)a and ranges from 0% to 18.1% in (7)b. Further, the gap of CF relative to E-MDSP ranges from 0% to 20.1% in (7)a and ranges from 0% to 20.5% in (7)b. The gap of FIC relative to E-MDSP ranges from 0% to 11.3% in (7)a and ranges from 0% to 10.3% in (7)b. We can see that only CF or only FIC can help to reduce the delay compared with S-MDSP. Combining CF and FIC together, which is E-MDSP, the delay is significantly reduced. In Fig. 7a, we observe that CF performs better than FIC at $|\text{hop}| = 2$, however this is not the case at other hop distances. Thus we cannot say that CF is better than FIC or not.

D. Transmission parallelization

We compare the two schemes with AMLS from the viewpoint of transmission parallelization, which is the average number of transmissions per compatible set used in a scheduling solution. For example, the average number of transmissions per compatible set for AMLS in Fig. 3a is $(2 + 2 + 2 + 2 + 2)/5 = 2$. We use N_L , N_I and N_{II} to represent the average number of transmissions per compatible set for AMLS, S-MDSP and E-MDSP respectively.

The results for networks of various sizes are shown in Fig. 8. In this figure, the number of packets equals 3, which are selected randomly among all node pairs with hop distance of 4. 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 S-MDSP. Thus optimizing delay is very different from maximizing spatial reuse of transmissions, demonstrating the significance of schemes tailored for delay-driven scheduling. However, E-MDSP is better than AMLS in terms of transmission parallelization. In E-MDSP, CF enhances the SINR by improving the numerator while FIC does so by reducing the denominator (as analyzed in Section 3.2 and Section 3.3), which enable more transmissions per compatible set.

7. 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. S-MDSP is a novel approach for delay-minimization scheduling, providing an efficient instrument to fully optimize the delay. E-MDSP 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. The work has also been partially supported by the Swedish Research Council grant 2013-05649. 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) [40]. A 3-SAT instance consists of B Boolean variables $\{v_1, v_2, \dots, v_B\}$ and C clauses $\{c_1, c_2, \dots, c_C\}$. For each variable v_i , $i = 1, 2, \dots, B$, we use \bar{v}_i to

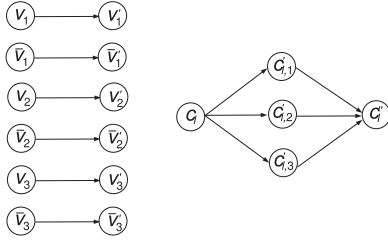


Fig. A1. An illustration of the literal links defined for three variables v_1, v_2, v_3 and their negations $\bar{v}_1, \bar{v}_2, \bar{v}_3$. The example clause c_1 is composed by $(v_1 \vee \bar{v}_2 \vee v_3)$. One compatible set is defined with respect to each literal link. For the example clause, link $(c_1, c'_{1,1})$ is compatible with (v_1, v'_1) , and $(c'_{1,1}, c'_1)$ is compatible with (\bar{v}_1, \bar{v}'_1) . Similar construction applies for the compatible sets defined for variable v_3, \bar{v}_3 , and clause c_1 . For v_2 and \bar{v}_2 , link $(c_1, c'_{1,2})$ is compatible with (\bar{v}_2, \bar{v}'_2) , and $(c'_{1,2}, c'_1)$ is compatible with (v_2, v'_2) .

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 $v_i \vee v_j \vee \bar{v}_k$. The decision problem is to determine whether or not there exists an assignment $Q : \{v_1, v_2, \dots, v_B\} \rightarrow \{\text{true}, \text{false}\}$ (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 \bar{v}_i , we define “literal” links (v_i, v'_i) , and (\bar{v}_i, \bar{v}'_i) , respectively. For every clause $c_l, l = 1, 2, \dots, C$, 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., \bar{v}_i , we use $(c_l, c'_{l,i})$ and $(c'_{l,i}, c'_l)$ 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 (\bar{v}_i, \bar{v}'_i) by \bar{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 \bar{v}_i . If the literal is a negation of a variable, say \bar{v}_k , then $(c_l, c'_{l,k})$ is added to compatible set \bar{v}_k , and $(c'_{l,k}, c'_l)$ 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 Fig. A.9.

Finally, we introduce a set S of $2B + C$ packets, referred to as “literal” and “clause” packets, respectively. The source and destination of a literal packet v_i ($i = 1, 2, \dots, B$) 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 $\bar{v}_i, i = 1, 2, \dots, B$. For a clause packet corresponding to $c_l, l = 1, 2, \dots, C$, 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 S-MDSP to deliver all packets to their destinations (i.e., D^*_S – 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 $D^*_S = 2B$ 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 \bar{v}_i for some $i = 1, 2, \dots, B$). 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 $v_i \vee v_j \vee \bar{v}_k$, then the delay of its packet will be greater than $2B$ if, and only if, $a(\bar{v}_i) < a(v_i)$, $a(\bar{v}_j) < a(v_j)$ and $a(v_k) < a(\bar{v}_k)$. 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 $2B + \min\{a(\bar{v}_i), a(\bar{v}_j), a(v_k)\}$). 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 \bar{v}_1 occupying the first two slots (in any order), compatible sets v_2 and \bar{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 $Q(v_i) = \text{true}$, $a(v_i) = 2i - 1, a(\bar{v}_i) = 2i$, and for each v_j with $Q(v_j) = \text{false}$, $a(v_j) = 2j, a(\bar{v}_j) = 2j - 1$. The crucial property of this correspondence is that a clause c_l ($i = 1, 2, \dots, C$) 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. \square

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 Netw. 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 Commun. Surv. Tutor. 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 Commun. Surv. Tutor. 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 SINR model, IEEE Trans. Wireless Commun. 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 Trans. Inf. Theory 60 (2) (2014) 1083–1100.
- [12] L. Badia, A. Botta, L. Lenzi, A genetic approach to joint routing and link scheduling for wireless mesh networks, Ad Hoc Netw. 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 Netw. 13 (0) (2014) 134–152.

- [14] M. Żotkiewicz, Max-Min fairness in WMNs with interference cancellation using overheard transmissions, *J. Appl. Math.* 2014 (2014).
- [15] L. Bui, R. Srikanth, 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 Trans. Netw.* 19 (1) (2011) 129–141.
- [17] B. Ji, C. Joo, N.B. Shroff, Delay-based back-pressure scheduling in multihop wireless networks, *IEEE/ACM Trans. Netw.* 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 Trans. Comput.* 64 (5) (2015) 1361–1374.
- [19] M. Cheng, Q. Ye, L. Cai, Cross-Layer schemes for reducing delay in multihop wireless networks, *IEEE Trans. Wireless Commun.* 12 (2) (2013) 928–937.
- [20] D. Gong, Y. Yang, Low-latency SINR-based data gathering in wireless sensor 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 Trans. Mob. Comput.* 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] Q. He, D. Yuan, A. Ephremides, Maximum link activation with cooperative transmission and interference cancellation in wireless networks, *IEEE Trans. Mob. Comput.* 16 (2) (2014) 408–421.
- [24] 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 Netw.* 9 (7) (2011) 1256–1269.
- [25] A. Sendonaris, E. Erkip, B. Aazhang, User cooperation diversity, part i: system description/part II: implementation aspects and performance analysis, *IEEE Trans. Commun.* 51 (11) (2003) 1927–1948.
- [26] X. Tao, X. Xu, Q. Cui, An overview of cooperative communications, *IEEE Commun. Mag.* 50 (6) (2012) 65–71.
- [27] S. Zhang, S. Liew, H. Wang, Blind known interference cancellation, *IEEE J. Sel. Areas Commun.* 31 (8) (2013) 1572–1582.
- [28] 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.
- [29] N. Michelusi, P. Popovski, O. Simeone, M. Levorato, M. Zorzi, Cognitive access policies under a primary ARQ process via forward-Backward interference cancellation, *IEEE J. Sel. Areas Commun.* 31 (11) (2013) 2374–2386.
- [30] P. Huang, L. Xiao, S. Soltani, M.W. Mutka, N. Xi, The evolution of MAC protocols in wireless sensor networks: A Survey, *IEEE Commun. Surv. Tutor.* 15 (1) (2013) 101–120.
- [31] H. Mostafaei, M. Menth, Software-defined wireless sensor networks: a survey, *J. Netw. Comput. Appl.* 119 (2018) 42–56.
- [32] E.C. van der Meulen, Three-terminal communication channels, *Adv. Appl. Probab.* 3 (1971) 120–154.
- [33] T.M. Cover, A.A.E. Gamal, Capacity theorems for the relay channel, *IEEE Trans. Inf. Theory* 25 (5) (1979) 572–584.
- [34] D. Yuan, V. Angelakis, L. Chen, E. Karipiditi, E.G. Larsson, On optimal link activation with interference cancellation in wireless networking, *IEEE Trans. Veh. Technol.* 62 (2) (2013) 939–945.
- [35] 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.
- [36] S. Gollakota, D. Katabi, Zigzag decoding: combating hidden terminals in wireless networks, in: *ACM SIGCOMM*, 2008, pp. 159–170.
- [37] X. Zhang, K.G. Shin, Chorus: collision resolution for efficient wireless broadcast, in: *IEEE INFOCOM*, Piscataway, NJ, USA, 2010, pp. 1747–1755.
- [38] M. Pióro, A. Tomaszewski, A. Capone, Maximization of multicast periodic traffic throughput in multi-hop wireless networks with broadcast transmissions, *Ad Hoc Netw.* 77 (2018) 119–142.
- [39] Y. Li, A. Capone, D. Yuan, On End-to-end delay minimization in wireless networks under the physical interference model, in: *IEEE INFOCOM*, 2015, pp. 2020–2028.
- [40] M. Garey, D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, USA, 1979.



Antonio Capone is Full Professor at Politecnico di Milano (Technical University of Milan), where he is Dean of the School of Industrial and Information Engineering, director of the Advanced Network Technologies Laboratory (ANTLab), and member of the strategic agenda team POLIMI 2040. His expertise is on networking and his main research activities include radio resource management in wireless networks, traffic management in software defined networks, network planning and optimization. On these topics he has published more than 250 peer-reviewed papers. He serves in the TPC of major conferences in networking, he is editor of *IEEE Trans. on Mobile Computing*, and *Computer Communications* (Elsevier), and he was editor of *ACM/IEEE Trans. on Networking and Computer Networks* (Elsevier). He is a fellow of the IEEE.



Yuan Li received his B.Sc. degree in computer science at National University of Defense Technology, China, and Ph.D. degree in network optimization at Lund University, Sweden, in 2008 and 2015, respectively. His research interests are in network modeling, algorithm design, integer programming and other combinatorial methods with applications in communication networks.



Michal Pióro is a professor and head of Computer Networks and Switching Division, Institute of Telecommunications, Warsaw University of Technology, Poland, and a professor at Lund University, Sweden. He received a Ph.D. degree in telecommunications in 1979 and the D.Sc. degree in 1990, both from the Warsaw University of Technology. In 2002 he received a Polish State Professorship. His research interests concentrate on modeling, design and performance evaluation of communication and computer networks. He is an author of four books (including well known monograph: M. Pióro and D. Medhi “Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan-Kaufmann, USA, 2004) and around 150 technical papers presented in communications and OR journals and conference proceedings. He has led many national and international research projects in the field for telecom industry and EC.



Di Yuan received the M.Sc. degree in computer science and engineering and the Ph.D. degree in optimization from the Linköping Institute of Technology in 1996 and 2001, respectively. In 2008, he joined the Politecnico di Milano, Italy, as a Guest Professor. In 2009 and 2012, he was a Senior Visiting Scientist with Ranplan Wireless Network Design Ltd, U.K. In 2011 and 2013, he held a part-time position with Ericsson Research, Sweden. In 2014 and 2015, he was a Visiting Professor with the University of Maryland, College Park, MD, USA. His current research mainly addresses network optimization of 4G and 5G systems and capacity optimization of wireless networks. He was a co-recipient of the IEEE ICC 2012 Best Paper Award. He was the supervisor of Dr. Lei Lei who received the Best Student Journal Paper Award by the IEEE Sweden Joint VT-COM-IT Chapter in 2014. He is an Area Editor of the *Computer Networks Journal*. He has been in the management committees of four European Cooperation in the Field of Scientific and Technical Research (COST) actions, invited lecturer of the European Network of Excellence EuroNF, and principal investigator of several European FP7 and Horizon 2020 projects.