Minimum latency link scheduling in TDMA wireless multi-hop networks

Paola Cappanera* Luciano Lenzini⋆ Alessandro Lori⋆ Gigliola Vaglini⋆

*Dipartimento di Sistemi e Informatica, Università di Firenze
Via S. Marta 3, 50139 Firenze - Italy

⋆Dipartimento di Ingegneria dell'Informazione, Università di Pisa
Via Diotisalvi 2, 56122 Pisa - Italy

Abstract

The link scheduling problem in Wireless Mesh Networks is addressed, which consists in determining the activation time of the links so as to minimize the end-to-end delay of origin-destination pairs traffic. Sink-tree topology network and constant bit rate traffics are considered. We propose a linear integer model where the non linear best delay bound known from the literature is approximated via a linear combination of the latencies experimented at each link of the network weighted according to some congestion criteria of the links. Two additional variants are presented which are based on the latency delay of origin-destination traffics. Preliminary computational results obtained on randomly generated instances via a off-the-shelf commercial solver are analyzed. More specifically, we observe that minimizing a suitable function of link latency seems to be a key issue in reducing the end-to-end delay of the network.

Keywords: *Wireless multi-hop networks, link scheduling, integer programming.*

1 Introduction

Wireless Mesh Networks (WMNs) are an emerging class of networks, usually built on fixed nodes that are inter-connected via wireless links to form a multi-hop network. Their main goal is to provide broadband access to mobile clients who are just on the edge of wired networks. WMNs can be used where cable deployment is not feasible or is too expensive, such as in remote valleys or rural areas, but also in offices and home environments. End-users are served by nodes called mesh routers, which are generally assumed to be stationary. Mesh routers are in turn wirelessly interconnected so as to form a network backhaul, where radio resource management challenges come into play. Moreover, some mesh routers are generally provided with access (e.g. through wires) to the Internet and therefore can act as gateways for the entire WMN. Communication between any two mesh routers as well as from any router to gateways is multi-hop.

Many of the WMN issues are thus common to those of multi-hop wireless networks, such as determining link scheduling in order to obtain high throughput efficiency or selecting appropriate routes between source and destination. However, the fact that mesh routers are fixed makes the backhaul of a WMN inherently different from distributed wireless networks (e.g. ad hoc networks), where the nodes may be portable devices. For example, problems such as energy consumption are no longer an issue. This makes it sensible to opt for a centralized network management, as opposed to the distributed approaches used for ad hoc wireless networks. In this case, nodes act in a coordinated fashion under the supervision of a network entity which determines the management based on global knowledge of the network topology and additional conditions.

The radio communication channel of WMNs (as of any other wireless networks) is broadcasting. Thus, one of the major problems experienced by WMNs is the reduction of capacity due to interference caused by simultaneous transmissions of nodes. One of the most widely used techniques to achieve robust and collision free communication is link scheduling operating in the context of time division multiplexing (TDM), according to which time is partitioned into slots of fixed duration and the links between neighbors compete for the slot resource.

A cross-layer approach where the routing and link scheduling functionalities are jointly addressed has been extensively studied in multi-hop wireless networks.

In the paper sink-tree WMNs are analyzed, i.e. networks for which routing is not an issue. The rationale behind this choice is that we want to focus on link scheduling and its impact on the end-to-end delay upper bound disregarding (at least for the moment) the routing functionality.

As usual, a commodity represents a flow of packets traversing a WMN between an origin node and a destination node. Due to the specific network topology (i.e. a sink tree) a commodity is thus identified by the unique path on the tree connecting its origin to its destination. In addition, each commodity accumulates a delay while traversing its path to the gateway. This delay can be regarded as made up of two components: the queuing delay and the scheduling delay. The former is due to flows of other commodities that have the same common route or that share a part of it, while the latter is due to the time spent at a node by each flow, waiting for the outbound link to be scheduled. For wired sink-tree networks this delay is bounded by a non-linear function of the latencies and rates of the network links [5], but a similar bound still holds for unwired networks. More specifically, with Constant Bit Rate (CBR) traffic the bound essentially depends on the link latencies. Expressing the latencies in terms of the scheduling variables, our aim is to define a model that minimizes the end-to-end delays of the commodities of the network.

In this paper we present a linear integer model for the link scheduling problem in TDM wireless mesh networks with a sink-tree topology and CBR traffic. Specifically, the non-linear delay bound defined in [5] is approximated via what we call the weighted latency delay (WLD) of a network, which is the sum of the latencies experienced at each link of the network, weighted according to some congestion criteria of the links. Also, we propose two variants of the previous model based on the latency delay of a commodity which is defined as the sum of the latencies experienced along the links of the corresponding path. In the former we minimize the maximum among the commodity latency delays, while in the latter the objective is to minimize the difference between the maximum and the minimum commodity delay. The three models are compared in terms of the bound on the actual delay accumulated by each commodity which is computed as in [5]. The solution of the scheduling problem can then be used to configure a network with delay bound guarantees, and the proposed models can be used both for network planning purposes or to implement real-time link schedulers.

2 Network model

We assume that slots are grouped into frames of N slots each, giving hence a frame duration of $T_f = NT_s$ seconds, where T_s is the slot duration measured in seconds and N is fixed by the network operator. The WMN is modeled by the *connectivity graph*, $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is a set of nodes representing the mesh routers and $E = \{e_1, \cdots, e_m\}$ is a set of directed links that connect nodes in the wireless range of each other.

In our model a set of sources want to send data to the root of a sink-tree graph; each source-sink pair for which there is a request is defined as a *commodity*. Each commodity $q \in Q$ has a *traffic demand* $r(q)$. The traffic demand is the bandwidth (bit/s) that commodity q should get to satisfy its Quality-of-Service (QoS). Each commodity's flow is considered constrained by a *leaky-bucket* shaper [5], with a *burst* σ and a *sustainable rate* ρ. We also assume the traffic as fluid.

We model the physical interference phenomenon occurring between the links of the wireless network following the well-known protocol interference models [1, 3, 4]. For each edge of the network $e \in E$ we define a conflicting set of edges $\mathcal{I}(e)$ which includes all the edges belonging to E which interfere with e ($\mathcal{I}(e)$ contains e itself); the interference condition is straightforwardly defined as follows:

$$\sum_{i \in \mathcal{I}(e)} x_i(t) \leq 1 \quad \text{if link } e \text{ is active in slot } t,$$

where $x_e(t)$ is a binary variable, such that $x_e(t) = 1$ if link $e \in E$ is active in slot t, and 0 otherwise. This means that if edge e is active in slot t, the associated interfering set $\mathcal{I}(e)$ must contain one active edge only (which is the edge e itself). Since a node, i.e. a mesh router, is equipped with a single Network Interface Card, which operates on a single frequency, its label can be used to address both the node and the upstream link leaving the node.

We translate the interference condition in a *conflict graph* $G_c = (E, C)$, where E is the set of links of the connectivity graph and $C = \{c_1, \ldots, c_r\}$ is the set of undirected edges that model conflicts within the network. Many types of conflicts can be modeled using this class of interference models, but in a

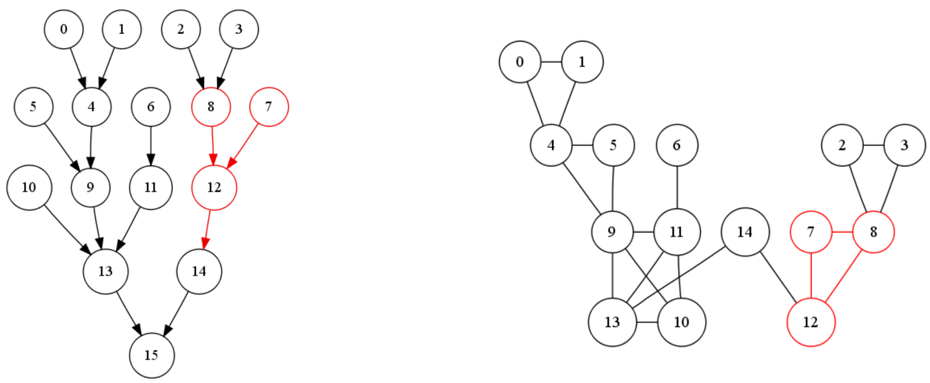

Figure 1: Conflicts in sink-tree TDMA networks

Time Division Multiple Access (TDMA) [6] network only a few types of conflicts are relevant [2]. Here we consider only direct neighbor conflicts, i.e. only those conflicts between links that share the same upstream node. For instance, given the network topology shown in Figure 1 (on the left), for edge 7 and 8 sharing the same upstream node, which is edge 12, we have the following edges in the conflict graph (on the right):

$$(7, 12), (8, 12), (7, 8).$$

Hence for each link e of the network, the set $\mathcal{I}(e)$ can be easily obtained retrieving the 1-hop neighborhood of node e in the conflict graph; in this example topology we have $\mathcal{I}(8) = \{2, 3, 7, 8, 12\}$. Given a conflict graph C, only those conflicts between *active links* have to be considered; an active link is a link with a not null flow to be scheduled. We thus define $C_f \subseteq C$ as the subset of conflicts involving active links:

$$C_f := \{(i, j) \in C : f_i > 0 \wedge f_j > 0\}$$

where f_i denotes the flow going trough link i.

In a similar way as in [2], we describe a schedule with a couple of vectors $\mathbf{\Pi} = [\pi_1, \ldots, \pi_m]^T$ and $\mathbf{\Delta} = [\Delta_1, \ldots, \Delta_m]^T$, representing respectively the activation time of each link of the network and the duration of its transmission. Since the time resource is slotted, these vectors are restricted to be non negative integer, i.e. $\mathbf{\Pi}, \mathbf{\Delta} \in \mathbb{Z}_+^m$. For a schedule to be valid it must also hold that:

$$\pi_i + \Delta_i \leq N \quad \forall i \in E,$$

which means that each link i must finish its transmission within the frame duration expressed in number of slots, i.e. N. The schedule must also ensure that the *conflict-free* condition is satisfied: while a link is transmitting, all of its conflicting links must retain to transmit until the link finishes its transmission. For any pair of active links i and j connected by and edge in the conflict graph we have:

- if j transmits after i, j should not transmit before i has completed the transmission

$$\pi_i - \pi_j + \Delta_i \leq 0$$

- changing the order of transmission we get

$$\pi_j - \pi_i + \Delta_j \leq 0.$$

The left-hand side of these two inequalities can be upper bounded by a constant equal to N since variables Π and Δ vary between 0 and N; introducing a binary variable o_{ij} (equal to 0 if i transmits before j) we can combine the two conditions linearizing the logic constraints. This completes the formulation of what we will refer to as the *conflict-free constrains*:

$$\begin{array}{ll} \pi_i - \pi_j + \Delta_i \leq N \cdot o_{ij} & \forall (i,j) \in C_f \\ \pi_j - \pi_i + \Delta_j \leq N \cdot (1 - o_{ij}) & \forall (i,j) \in C_f \\ \pi_i + \Delta_i \leq N & \forall i \in E. \end{array}$$

These constraints are necessary and sufficient conditions for a link schedule to be conflict-free and must hold for each pair of active conflicting links of the network.

3 The minimum latency scheduling problem

We now introduce a {0,1}-integer program to find a feasible conflict-free schedule that minimizes the end-to-end delay of the commodities in the network. For each commodity $q \in Q$, we denote with $P(q)$ the set that contains the edges of the traversed path. We assume that the routing in the network is given as a vector $\mathbf{f}$ of rates (bit/s) such that

$$f_e = \sum_{q \in Q} x_e^q \qquad \forall e \in E,$$

where x_e^q is the traffic rate of commodity q on link e. Given the capacity c_e of each link $e \in E$ (bit/s) and the frame duration N (slot), we force the link to transmit at least a fraction of the frame equals to the utilization of the link capacity:

$$\Delta_e \geq N \cdot \frac{f_e}{c_e} \qquad \forall e \in E.$$

In case of a sink-tree topology and CBR traffic the non-linear delay on a commodity path strictly depends on the latency, which is a linear function of the scheduling variables Δ_e, i.e. the transmission duration of links $e \in P(q)$ expressed in number of slots. For a link $e \in E$ the latency is $(N - \Delta_e) \cdot T_s$ and represents the time interval between two consecutive activations of link e.

We want to minimize the weighted sum of the latencies of the links, where the link weight is given by summing up the commodity rates which traverse that link:

$$w_e = \sum_{q \in Q_e} \rho_q \qquad \forall e \in E,$$

where Q_e is the set containing the indices of commodities traversing link e.
Given a graph G and an assignment for Δ, we define the *weighted latency* of a network as follows:

$$\mathrm{WL}(G, \Delta) = \sum_{e \in E} w_e (N - \Delta_e) \cdot T_S.$$

Therefore we write the minimum weighted latency scheduling (MinWL) problem as

$$\min \quad \mathrm{WL}(G, \Delta)$$

$$
\begin{aligned}
\text{subject to}: \quad & \pi_i - \pi_j + \Delta_i \leq N \cdot o_{ij} && \forall (i,j) \in C_f && (1)\\
& \pi_j - \pi_i + \Delta_j \leq N \cdot (1 - o_{ij}) && \forall (i,j) \in C_f && (2)\\
& \pi_e + \Delta_e \leq N && \forall e \in E && (3)\\
& N \cdot \frac{f_e}{c_e} \leq \Delta_e && \forall e \in E && (4)\\
& \pi_e, \Delta_e \in [0, .., N] && \forall e \in E && (5)\\
& o_{ij} \in \{0, 1\} && \forall (i,j) \in C_f && (6)
\end{aligned}
$$

In case of commodities with homogeneous rates, w_e can be set to the number of commodities traversing link e. This setting corresponds to the normalization of weights w_e defined above with respect to the common bitrate of the commodities. But it also holds that

$$\mathrm{WL}(G, \Delta) = \sum_{e \in E} w_e (N - \Delta_e) \cdot T_s = \sum_{e \in E} |Q_e| (N - \Delta_e) \cdot T_s = \sum_{q \in Q} \sum_{e \in P(q)} (N - \Delta_e) \cdot T_s,$$

that is the sum over all commodities of what we will call the *commodity latency delay*:

$$D_q = \sum_{e \in P(q)} (N - \Delta_e) \cdot T_s.$$

For this special case we then write the model as

$$\min\Big\{\sum_{q \in Q} D_q \quad \text{s. t.:} \quad (1), (2), (3), (4), (5), (6)\Big\}.$$

Finally we derive two variants of this model which, without loss of generality, are presented here for the homogeneous rate case; in the first one, which we call the min-max latency delay (MinMaxLD) problem, we minimize the maximum delay among commodity delays:

$$\min\Big\{D_{Max} \quad \text{s. t.:} \quad \sum_{e \in P(q)} (N - \Delta_e) \leq D_{Max} \; \forall q \in Q, \;\; (1), (2), (3), (4), (5), (6)\Big\}.$$

The last formulation (MinDiffLD) minimizes the difference between the maximum and the minimum among the commodity delays:

$$\min\Big\{D_{Max} - D_{Min} \quad \text{s. t.:} \quad D_{Min} \leq \sum_{e \in P(q)} (N - \Delta_e) \leq D_{Max} \; \forall q \in Q, \;\; (1), (2), (3), (4), (5), (6)\Big\}.$$

4 Computational results

We tested the proposed models on several randomly generated instances with a sink-tree topology, including random tree and balanced binary tree networks; we also solved instances with all-leaves-to-sink and all-nodes-to-sink commodities, both with homogeneous and heterogeneous rates requirements. For each instance we solved optimally the three proposed models via the mixed integer programming solver ILOG CPLEX 11. The solutions thus obtained are given in input to the framework introduced in [5] with the aim to get the relative delay upper bounds. Finally we compared the models in terms of minimum, maximum and average delay bounds.

Here we present preliminary computational results obtained on 5 randomly generated instances for the homogeneous case; we set parameters N and T_s respectively to 100 and 10^{-5}. Tables report for each

Model	$\mathbf{D}_{\min}$(msec)	$\mathbf{D}_{\max}$	$\mathbf{D}_{\mathbf{ave}}$	**CPU time (s)**	**B&B nodes**	**Cols**	**Rows**
MinWL	1.618	8.516	5.622	0.012	54	19	20
MinMaxLD	2.517	7.308	5.711	0.020	38	20	26
MinDiffLD	3.418	7.694	6.269	0.012	26	21	32
TDMA-Delay	3.420	9.172	7.254	-	-	-	-

Table 1: 7 nodes balanced binary tree, 1000 Kb/s rate, all-nodes-to-sink

Model	$\mathbf{D}_{\min}$(msec)	$\mathbf{D}_{\max}$	$\mathbf{D}_{\mathbf{ave}}$	**CPU time (s)**	**B&B nodes**	**Cols**	**Rows**
MinWL	2.165	14.067	9.117	0.72	4111	48	66
MinMaxLD	2.513	11.059	8.613	0.52	1661	49	80
MinDiffLD	4.267	16.803	13.353	1.18	5522	50	94
TDMA-Delay	4.267	19.455	15.465	-	-	-	-

Table 2: 15 nodes balanced binary tree, 200 Kb/s rate, all-nodes-to-sink

Model	$\mathbf{D}_{\min}$(msec)	$\mathbf{D}_{\max}$	$\mathbf{D}_{\mathbf{ave}}$	**CPU time (s)**	**B&B nodes**	**Cols**	**Rows**
MinWL	4.087	8.684	6.334	0.108	430	47	52
MinMaxLD	7.992	8.905	8.394	0.012	6	48	57
MinDiffLD	7.992	8.905	8.394	0.016	6	49	62
TDMA-Delay	8.008	15.952	11.37	-	-	-	-

Table 3: 15 nodes random path length star, 1000 Kb/s rate, all-leaves-to-sink

Model	$\mathbf{D}_{\min}$(msec)	$\mathbf{D}_{\max}$	$\mathbf{D}_{\mathbf{ave}}$	**CPU time (s)**	**B&B nodes**	**Cols**	**Rows**
MinWL	1.219	15.778	9.598	1.744	5227	48	54
MinMaxLD	2.317	13.476	9.304	0.352	1068	49	68
MinDiffLD	3.768	16.225	11.302	0.408	1272	50	82
TDMA-Delay	4.123	23.536	16.071	-	-	-	-

Table 4: 15 nodes random tree, 300 Kb/s rate, all-nodes-to-sink

Model	$\mathbf{D}_{\min}$(msec)	$\mathbf{D}_{\max}$	$\mathbf{D}_{\mathbf{ave}}$	**CPU time (s)**	**B&B nodes**	**Cols**	**Rows**
MinWL	1.013	23.999	14.385	11751	43674695	107	127
MinMaxLD	1.013	24.280	16.107	2.916	7847	108	156
MinDiffLD	4.014	27.696	19.083	5.608	16323	109	185
TDMA-Delay	4.014	40.125	24.763	-	-	-	-

Table 5: 30 nodes random tree, 100 Kb/s rate, all-nodes-to-sink

test instance: minimum, maximum and average commodity delay, solver solution time, Branch & Bound nodes, number of variables and constraints.

We observe the following facts: (*i*) minimize a proper function of the link latencies seems to be crucial when the focus is on the end-to-end delay of the commodities. Whatever of the three proposed models is used we observe a remarkable reduction of the end-to-end delay bounds (see column $D_{\max}$) with respect to the state-of-the-art scheduling algorithms which concentrate on throughput maximization or TDMA delay. Such algorithms do not operate on link rates and only the rates requested by the routing phase are scheduled; (*ii*) the reduction of the maximum commodity delay with respect to the one experimented with a TDMA approach increases as the dimension of the network increases and as the regularity of the network is lost (balanced binary tree vs random tree topology); (*iii*) in case of homogeneous rate commodities and balanced binary tree topology (see Tables 1 and 2) model MinMaxLD performs better than model

MinWL; such results seem to show that, at least for this class of instances, a min-max objective is able to approximate well the actual delay whose bound is given by a non-linear combination of link latencies; (*iv*) in case of homogeneous rate and random tree topology discording results are reported (see Tables 3, 4 and 5) i.e. no model outperforms the others. A more in depth analysis is required to get insights on which kind of model captures the non-linear behavior of the closed-form proposed in [5] the most; (*v*) model MinDiffLD should be further investigated: minimizing the difference between the maximum and the minimum commodity delay might offer the opportunity to increase the utilization factor of the links which is a highly desirable characteristic in setting of this type. In that case preliminary results seem to show that the set of commodities to evaluate in the objective function should be accurately chosen in order to prevent undesirable effects from occurring. For example should the two paths giving the maximum and the minimum delay be one a subpath of the other, the utilization factor of the common links might be quite low; on the contrary it seems reasonable to assume that bigger rates on those common links might help in reducing the bottleneck effects.

Finally we conclude that the computational results shown in this work, even if still preliminary, are quite encouraging and the study of models which approximate well the non-linear delay bound given in [5] seems to identify a promising line of research. A deeper analysis of the models is required: specifically the models should be extended so as to work also on more general network topologies, with non-CBR traffic and considering downstream traffic as well. The design of ad-hoc real-time algorithms to solve these models is an additional line of research we are pursuing.

References

[1] L. Badia, A. Erta, L. Lenzini, and M. Zorzi. Scheduling, routing, and related cross-layer management through link activation procedures in wireless mesh networks. In E. Hossain and K. K. Leung, editors, *Wireless Mesh Networks, Architectures and Protocols*. Springer, 2008.

[2] P. Djukic and S. Valaee. Delay aware link scheduling for multi-hop TDMA wireless networks. *Networking, IEEE/ACM Transactions on*, 2009.

[3] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, Mar 2000.

[4] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels. *Networking, IEEE/ACM Transactions on*, 13(4):868–880, Aug. 2005.

[5] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea. Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks. *Performance Evaluation*, 63(9-10):956–987, October 2006.

[6] R. Nelson and L. Kleinrock. Spatial TDMA: a collision-free multihop channel access protocol. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 33(9):934–944, Sep 1985.