# Quality-of-Service Provisioning for Multi-Service TDMA Mesh Networks

Petar Djukic and Shahrokh Valaee[1] *

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
10 King's College Road, Toronto, ON, M5S 3G4, Canada
{djukic,valaee}@comm.utoronto.ca

**Abstract.** Multi-service mesh networks allow existence of guaranteed delay Quality-of-Service (QoS) traffic streams such as Voice over IP and best effort QoS traffic streams such as file transfer. We present an optimization that performs a linear search for the minimum number of TDMA slots required to support the guaranteed QoS flows. At each stage of the search a linear integer program is solved to find if there is a feasible schedule supporting the required end-to-end bandwidth and delay. Our optimization results in a relative order of transmissions in the frame that guarantees a maximum end-to-end delay in the network. The ordering of the transmissions can be used later to find feasible schedules with the Bellman-Ford algorithm on the conflict graph for the network. We use the optimization in numerical simulations showing the efficiency of 802.16 mesh networks with VoIP traffic.

## 1 Introduction

Wireless mesh networks interconnect access points (APs) spread out over a large geographical area. Wireless terminals (WTs) connect to the access points on their first hop and their traffic is carried by the wireless mesh to the Point-of-Presence (POP) where it can go to the Internet. The POP is the only node in the network connected to the Internet and can also act as a base station (mesh coordinator). Current mesh networks use 802.11 technology to interconnect the mesh backbone [1, 2]. However, 802.11 technology is a decade old and was not designed for mesh networks. In particular, 802.11 lacks the extensions to provide Quality–of–Service (QoS) in multihop wireless environments [3].

New mesh network technologies such as 802.16 (WiMax) and 802.11s are designed to provide QoS with Time Division Multiple Access (TDMA) [4, 5]. In TDMA, end-to-end QoS is negotiated in terms of end-to-end bandwidth reserved for each AP on the links connecting it to the POP. QoS is enforced at each link with scheduled access to the wireless channel. A schedule assigns slots from each TDMA frame to links so that a number of non-conflicting links can transmit simultaneously in each slot. Link bandwidth is given by the number of slots assigned to it in each frame and the modulation used in the slots.

* This work was sponsored in part by the LG Electronics Corporation.

If all slots in a frame are reserved, TDMA mesh networks would not allow statistical multiplexing of best effort data streams at the MAC layer. So, both 802.16 and 802.11s divide the slots in every frame between guaranteed service traffic streams and best effort traffic streams. In 802.16, slots reserved for guaranteed QoS traffic are assigned with the centralized scheduling protocol, while other slots are assigned with the decentralized scheduling protocol. In the centralized scheduling protocol, the mesh coordinator assigns bandwidth to all links in the network based on traffic demands from the APs. On the other hand, in the decentralized scheduling protocol, mesh routers are free to negotiate pairwise TDMA assignments, with no QoS guarantees on the bandwidth. In 802.11s networks, slots dedicated to TDMA access are negotiated in a pairwise fashion, while the rest of the frame is dedicated to best effort service with 802.11 EDCF.

In this paper we answer the following QoS provisioning question: *What is the minimum number of TDMA slots required to support a required guaranteed QoS in the network?* The QoS is specified both in terms of bandwidth and delay, for traffic streams such as Voice over Internet Protocol (VoIP). The end-to-end bandwidth of each AP is guaranteed with a TDMA schedule that assigns the appropriate bandwidth to links connecting the AP to the POP. The delay in the network consists of queueing delay due to traffic variations and TDMA propagation delay. TDMA propagation delay occurs when an outgoing link on a mesh node is scheduled to transmit before an incoming link in the path of a packet [6]. In this paper, we assume that the queueing delay is minimized in the network layer with the assignment of link bandwidths and concentrate on scheduling algorithms that guarantee a bound on the TDMA delay. We have shown in [6] that end-to-end TDMA propagation delay accumulates at each hop and can be very large – multiples of TDMA frame duration. In 802.16 frame duration can be as large as 20ms, so TDMA delay can be relatively large compared to the 150ms delay budget required for VoIP quality [7].

We formulate an optimization that minimizes the number of TDMA slots allocated for guaranteed QoS traffic, subject to the constraint that the delay introduced with TDMA scheduling is bellow a given threshold. The bound on TDMA propagation delay can be found by delay budgeting the network across the mesh and the wired backbones. The maximum allowed TDMA delay is found by subtracting the delay due to voice processing and the jitter buffer delay from the overall delay budget [7]. The optimization performs a linear search for the minimum number of TDMA slots. At each step of the search, the optimization increases the number of slots required for guaranteed QoS and solves an integer program that finds a transmission order that supports the required banwidth at each hop, subject to the TDMA delay. The optimization stops as soon as the number of guaranteed slots with a feasible transmission schedule is found. We have shown that with a known transmission order TDMA schedules can be found with the Bellman-Ford algorithm run on the conflict graph for the wireless network [6,8]. Since end-to-end TDMA delay depends on the transmission order alone, it can be distributed to the nodes as a part of their QoS provisioning information, thus making sure that the resulting TDMA schedules have a fixed

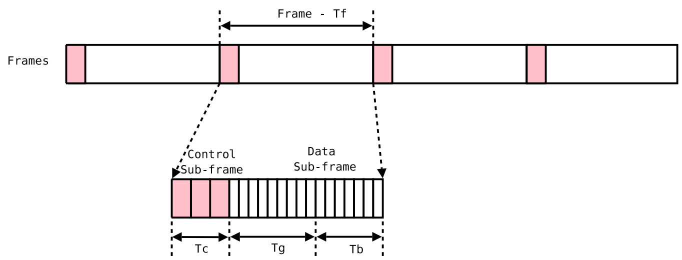



**Fig. 1.** Multi-service TDMA frames

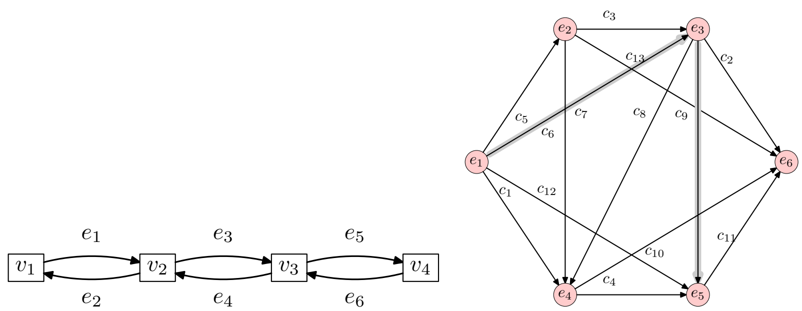
maximum TDMA delay. Using this method, schedules can be changed dynamically when the bandwidth changes, but still maintain the maximum end-to-end delay.

Our optimization is suitable for mesh network planning. During the planning process, locations of mesh nodes are chosen based on the expected interference from neighbouring nodes [4]. With the location known, the expected interference is also known, making it possible to predict maximum modulation at each link, as well as the mesh topology. Given the network topology and the maximum bit-rate on each link, it is possible to plan end-to-end bandwidth to support a specified number of VoIP connections. Our optimization adds an additional level of QoS planning for mesh networks – the scheduling delay through the mesh.

We study the planing for 802.16 mesh networks with numerical simulations. We examine the effect of 802.16 frame size on efficiency of carrying VoIP traffic. Efficiency is defined as the number of slots required by guaranteed QoS traffic divided by the total number of TDMA slots in the frame. We show that increasing the frame size increases the efficiency in the network almost 50% for a low number of VoIP calls, however for a high number of VoIP calls the increase in the efficiency is less than 5%. The efficiency increases with the frame size because as the frame size increases, more transmission orders can produce TDMA schedules with the required bandwidth requirement.

## 2 Network and Transmission Model

The mesh network is using a time division multiple access (TDMA) MAC protocol [4]. In TDMA MAC protocols, the time is divided into slots of fixed duration, which are then grouped into frames. A fixed portion of each frame is dedicated to control traffic, while the rest of the slots are used for data traffic. Each frame consists of $N_f$ slots, where $N_c$ of the slots are allocated for control traffic, $N_g$ are reserved for guaranteed service data traffic and $N_b$ are reserved for best effort data traffic (Fig. 1). Frames have duration of $T_f = N_f T_S$ seconds, where $T_S$ is the slot duration. In this paper, we minimize the number of slots reserved for guaranteed service traffic, $N_g$.




(a) A four node chain

(b) Conflict graph for the four node topology

**Fig. 2.** Chain topology and its conflict graph.

We model the mesh with a topology graph connecting wireless routers in the range of each other. We assume that if two routers are in the range of each other, they establish links in the MAC layer, so the TDMA network can be represented with a connectivity graph $G(V, E, f_t)$, where $V = \{v_1, \ldots, v_n\}$ is the set of nodes[1], $E = \{e_1, \ldots, e_m\}$ are directional links between neighbouring nodes, and $f_t : E \to V \times V$ assigns links to pairs of nodes. Links are directional, so for a link $e_k \in E$, $f_t(e_k) = (v_i, v_j)$ means that traffic on the link is transmitted from $v_i$ to $v_j$. The links operate at different bitrates, which depend on the signal-to-noise ratio. Signal-to-noise ratio is divided into several discrete levels and each is associated with its corresponding bitrate. We define the link bitrate as the number of bits transmitted in a TDMA slot, represented with the mapping $b : E \to \{M_1, M_2, \ldots, M_{\max}\}$, where $M_1$ is the number of data bits carried in a slot with the minimum modulation and $M_{\max}$ is the number of bits carried in a slot with the maximum modulation and coding.

We assume that the signal-to-noise ratio of each link depends on the wireless channel alone and not other links in the network, meaning that competing links do not transmit at the same time. Under this model of transmission, in TDMA networks, a receiver can only have one active link at any given time. In a single hop neighbourhood, this means all links interfere with each other. In a two hop neighbourhood, two links, whose nodes are two hops away, interfere if the receiver of one of the links is in the transmission range of the other link.

We keep track of conflicts between the links with conflict graphs. Conflict graphs can be defined with a triplet $G_c(E, C, f_c)$, where $E$ is the set of links, $C = \{c_1, \ldots c_r\}$ is the set of TDMA conflicts, one for each of the $r$ conflicting pairs of links, and $f_c : C \to \{\{e_i, e_j\}, e_i, e_j \in E\}$ associates the conflicts with pairs of links.[2] The graph is undirected since conflicts are symmetrical. In this paper,

[1] We use the convention that $v_n$ is the POP.

[2] We use the notation $\{\cdot\}$ for unordered sets and $(\cdot)$ for ordered sets, so $f_c$ defines an undirected graph.

we use a conflict graph with an arbitrary assignment of directions to the arcs, $\overrightarrow{G}_c(E, C, \overrightarrow{f}_c)$, where $\overrightarrow{f}_c : C \rightarrow E \times E$. The directed conflict graph simplifies the derivation of formulas, however, as we have shown in [6], the arbitrary orientation of arcs does not cause any loss of generality. We use the four node example from Fig. 2a to demonstrate how the arcs in the conflict graph are created. The vertices in the conflict graph are the six links from the topology graph. All of the links conflict with each other, except for pairs $e_1$ and $e_6$ and $e_2$ and $e_5$, so they are not connected in the conflict graph (Fig. 2b). The orientation of the conflicts was chosen randomly, since it does not change the resulting delay or wireless capacity [6].

Link bandwidths are assigned so that a certain number of VoIP connections can be carried between each AP and the POP. The assignment of bandwidths is performed during mesh network planning. The assignment of link bandwidths is provided as the mapping $R : E \rightarrow \mathbb{R}_{[0,\infty)}$. The scheduling algorithm assigns link bandwith through the number of slots a link can use in a frame $d : E \rightarrow \mathbb{Z}_{[0,T]}$. The number of slots required to achieve bandwidth $R_i$ on link $e_i$ can be found with:

$$d_i = \left\lceil \frac{R_i T_f}{b_i} \right\rceil = \left\lceil \frac{R_i N_f T_S}{b_i} \right\rceil, \tag{1}$$

where $\lceil \cdot \rceil$ denotes the ceiling of a real number, $T_f$ is the duration of the frame, $N_f$ is the number of slots in the frame, $T_s$ is the duration of mini-slot in seconds and $b_i$ is the number of bits in each slot.

We assume that after the link bandwidths have been assigned, there are $2q$ one-way paths terminating or originating at the POP. The paths connect the POP with $q < n - 1$ APs acting as VoIP cells for WTs. We denote a path from the POP to node $v_l$ with $\mathcal{P}_l$ and the path from the node $v_l$ to the POP with $\mathcal{P}_{q+l}$. The set of all paths is denoted with $P = \{\mathcal{P}_1, \ldots, \mathcal{P}_{2q}\}$. We use a mapping function $\overrightarrow{f}_{\mathcal{P}} : P \rightarrow E \times E$ to associate a path with its starting and ending links, so $\overrightarrow{f}_{\mathcal{P}}(\mathcal{P}_l) = (e_i, e_j)$ means the link for the first hop is $e_i$ and the link for the last hop is $e_j$.

## 3 TDMA Scheduling

We present a set of conditions that guarantees that the transmission schedule for guaranteed QoS slots is both *valid* and *conflict-free*. A valid transmission schedule assigns the number of slots allocated to the links due to QoS requirements. A conflict-free schedule ensures that transmissions of conflicting links do not overlap. These conditions are used in the minimization of $N_g$ as constraints, to ensure that a given $N_g$ results in a feasible schedule for the guaranteed QoS slots. A transmission schedule assigns slots from each TDMA frame to links so that a number of non-conflicting links can transmit simultaneously during each slot.

We define the TDMA schedule, used for guaranteed QoS service slots, with a pair of vectors $\boldsymbol{\pi}$ and $\boldsymbol{d}$, where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_m]^T$ is the vector of link starting

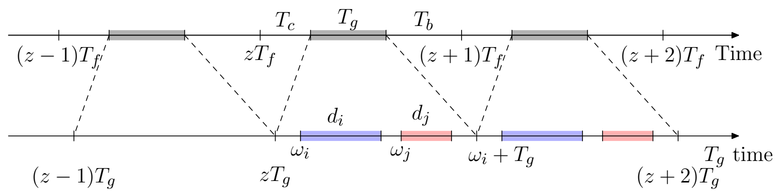



**Fig. 3.** Conflict-free Conditions

times in the part of the frame allocated for guaranteed QoS and $\boldsymbol{d} = [d_1, \ldots, d_m]^T$ is the vector of the number of slots each link transmits in the frame. The activation times need to be limited to $\pi_i \in [0, N_g)$, $\forall e_i \in E$, so that each link transmits in every frame. If we assume that the slots in the frame are numbered $[0, N_f - 1]$, the transmission takes place in slot $N_c + \pi_i$ with the duration of $d_i$ slots. We note that the schedule is valid by definition, since every link will be scheduled to transmit for the number of slots required with its bandwidth assignment. We have defined conditions for conflict-free scheduling in [6]; we briefly summarize those results here.

The guaranteed TDMA schedule repeats in every frame, until a new set of link bandwidths is assigned. The slots allocated for guaranteed QoS are always sandwitched between the control slots and the best effort TDMA slots. If we ignore the non-guaranteed QoS slots, we can view the uninterrupted sequence of guaranteed QoS slots on its own axis (Fig. 3). On this axis, the activation times, $\boldsymbol{\pi}$, are periodic. Periodicity of the schedule means that the start time $\pi_i$ for link $e_i$ actually represents a series of activation times, which can be derived from $\pi_i$ by adding multiples of $N_g$ slots (Fig. 3). We denote with $\Pi_i = \{\pi_i + z_i N_g, z_i \in \mathbb{Z}\}$ the series of activation times for link $e_i$, generated with $\pi_i$. The actual activation time in the frame $\pi_i$ can be found from any activation time $\omega_i \in \Pi_i$ with the modulo operator: $\pi_i = \omega_i \pmod{N_g}$.

The number of times a link transmits in the frame depends on its starting time and the duration of its transmission. If for some link $e_i$, $\pi_i + d_i \leq N_g$, the link will transmit once per frame. On the other hand, if $\pi_i + d_i > N_g$, the link will be scheduled twice for transmission in the frame. The first transmission starts in slot $N_c$, with the duration $\pi_i + d_i - N_g$ slots and the second transmission starts in slot $N_c + \pi_i$ with the duration of $N_g - \pi_i$ slots. So, our scheme limits the number of transmissions by any link to at most two in a frame. This is good for protocols such as 802.16 where the overhead of each transmission can be as much as 324 bytes at the highest modulation [4]. In [6], we also show how this method can be extended to find schedules for multiple activation times in the guaranteed QoS part of the frame.

The conflict-free conditions for a schedule can be expressed in terms of points in the sequences $\Pi_i$, $\forall e_i \in E$. We have shown in [6] that a schedule is conflict-free, if for any two conflicting links $e_i$ and $e_j$ whose conflict is $c_k \in C : \overrightarrow{f}_c(c_k) =$

$(e_i, e_j)$:

$$d_i \leq \omega_j - \omega_i + p_k N_g \leq N_g - d_j, \tag{2}$$

where $\omega_i \in \Pi_i$ and $\omega_j \in \Pi_j$ and $p_k = 0$ if $\omega_j - \omega_i > 0$ and $p_k = 1$ if $\omega_j - \omega_i < 0$. Variable $p_k$ specifies a relative order of transmissions, which prompts us to refer to it as the "transmission order" in the rest of the paper. A schedule is conflict free if (2) is true for all conflicts in the network. Fig. 3 shows why (2) is necessary for the schedule to be conflict-free. In the figure, $p_k = 0$, so we are comparing the timing of $e_i$'s transmission to the first transmission of $e_j$ that follows it. Clearly, it is necessary that $\omega_j \geq \omega_i + d_i$ since $e_j$ cannot start its transmission before $e_i$ finishes. Also, the next transmission of $e_i$ should be after $e_j$ has finished its transmission, so $\omega_i + n_g \geq \omega_j + d_j$. Full proof of necessity and sufficiency of (2) can be found in [6].

We show next that the TDMA delay depends on the transmission order and a feasible $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_m]^T$. However, we also show that the feasible $\boldsymbol{\omega}$ can be compressed into a single parameter, leading us to a two step procedure to optimize TDMA delay. First, TDMA propagation delay is minimized subject to an existence of a feasible schedule. Second, the transmission order and the feasibility parameter are distributed among the mesh nodes, so they can find the transmission schedule using the Bellman-Ford algorithm.

## 4 TDMA Delay

We show how to calculate and minimize return trip TDMA propagation delay in the mesh in [6]. While that approach is appropriate for TCP flows, it is not appropriate for VoIP connections, since perceived voice quality depends on the *one-way* delay between a sender and its receiver [7]. In this section, we find the expression for one way TDMA propagation delay on a path. We first find TDMA propagation delay at single router on the path and then add up the delay at every router on the path to find an expression for the one-way end-to-end TDMA delay on the path.

TDMA propagation delay occurs if an ingress link is scheduled to transmit after an egress link on the router. So, on a single mesh router it is measured as the time between the transmission of an ingress link, to the time when the egress link transmits, excluding the queueing delay. We note that the TDMA propagation delay experienced by a packet traversing a mesh router from an ingress link $e_i$ to an egress link $e_j$, in slots, is given by:

$$\Delta_k = \begin{cases} \omega_j - \omega_i + p_k N_f & \text{if } \overrightarrow{f}_c(c_k) = (e_i, e_j) \\ \omega_j - \omega_i + (1 - p_k) N_f & \text{if } \overrightarrow{f}_c(c_k) = (e_j, e_i), \end{cases} \tag{3}$$

where $c_k \in C$ is the conflict connecting the two links in the conflict graph and $\omega_i$, $\omega_j$ and $p_k$ correspond to a fixed feasible schedule $S(\boldsymbol{\pi}, \boldsymbol{d})$. The delay in seconds can be found by multiplying $\Delta_k$ with the slot duration $T_S$. For example, if $\overrightarrow{f}_c(c_k) = (e_i, e_j)$ and $p_k = 0$ it is easy to see that $\Delta_k = \omega_j - \omega_i$ since the packet can be transmitted in the same frame on both links. However, if $p_k = 1$, $\Delta_k = \omega_j - \omega_i + N_f$ since the packet has to wait for new frame to be transmitted by $e_j$.

The total TDMA delay on a path is found by adding up the delay at each router on the path in the topology graph. We now show that each path in the topology graph corresponds to a path in the conflict graph, which lead us to a simpler formulation of the TDMA delay. The path in the conflict graph, corresponding to a path in the topology graph, can be obtained by traversing the conflicts in $\overrightarrow{G}_c(V, E, \overrightarrow{f}_c)$ corresponding to conflicts between ingress and egress links at each router in the path. For example, path $e_1 \rightsquigarrow e_3 \rightsquigarrow e_5$ in the four node topology shown in Fig. 2a, corresponds to the path $c_6 \rightsquigarrow c_9$ in Fig. 2b. We represent the paths in the conflict graph with $r$-sized vectors in the $\{-1, 0, 1\}^r$ path space of the conflict graph [9]. The meaning of the entries of $\boldsymbol{\theta}_l = [\theta_1, \ldots, \theta_r]^T$, corresponding to path $\mathcal{P}_l$ in the conflict graph, is:

$$\forall c_k \in C, \quad \theta_k = \begin{cases} 1, & \text{if } c_k \in \boldsymbol{\theta}_l^+ \\ -1, & \text{if } c_k \in \boldsymbol{\theta}_l^+ \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $\boldsymbol{\theta}_l^+$ is the set of arcs in the positive direction of $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_l^+$ is the set of arcs in the negative direction of $\boldsymbol{\theta}_l^+$. For example, the path emphasized in Fig. 2 corresponds to the vector $\boldsymbol{\theta} = [0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0]^T$.

The total delay on path $\mathcal{P}_l$ is found by adding up the single hop delay incurred for the conflicts between ingress and egress links at each router in the path. The delay on path $\mathcal{P}_l$ is given by:

$$D(\mathcal{P}_l) = \sum_{k=1}^{r} \theta_k \Big(\tau_k + p_k N_f\Big) I(\theta_k > 0) + \sum_{k=1}^{r} \theta_k \Big(\tau_k + p_k N_f - N_f\Big) I(\theta_k < 0) \tag{5}$$

where $\tau_k = \omega_j - \omega_i$ is the tension for the conflict $c_k$, $\overrightarrow{f}_c(c_k) = (e_i, e_j)$, and $I(\cdot)$ is the indicator function, that is 0 when its argument is false and 1 when its argument is true. A well known property of tensions is that the sum of tensions along a path is equal to the tension between end vertices [9]. This property allows us to express the delay on the path with:

$$D(\mathcal{P}_l) = \omega_j - \omega_i + \boldsymbol{\theta}_l^T \boldsymbol{p} N_f + \sum_{k=1}^{r} N_f I(\theta_k < 0), \tag{6}$$

where $e_i$ and $e_j$ are the first and the last link on the path, $\overrightarrow{f}_{\mathcal{P}}(\mathcal{P}_l) = (e_i, e_j)$, and we have used vector product to express the summation of $\theta_k p_k$ on the path. Since the last term in the delay is a constant depending only on the orientation in the conflict graph, we will refer to it with $D_l = \sum_{k=1}^{r} N_f I(\theta_k < 0)$ for path $\mathcal{P}_l$ in the rest of the paper.[3]

[3] $D_l$ depends on the orientation of the conflict graph. However, we show in [6] that since $\boldsymbol{p}$ also depends on the orientation of the conflict graph, the total TDMA propagation delay does not change if the orientation changes.

## 5 QoS Provisioning for Minimum Delay

In this section, we present an algorithm that can be used to find the minimum number of guaranteed QoS slots, required to support a given bandwidth subject to maximum TDMA delay. The maximum TDMA delay in slots is found with delay budgeting and is denoted with $N_{\max} = D_{\max}/T_S$, where $D_{\max}$ is the maximum allowed delay and $T_S$ is the slot duration. We present the algorithm first and then show how to compress a feasible schedule associated with the minimum $N_g$ into two provisioning parameters that should be distributed to all mesh routers.

The minimum $N_g$ can be found with a non-linear $\{0, 1\}$-integer program. However, we simplify this optimization by finding the minimum $N_g$ with a search algorithm. The algorithms starts with $N_g = 1$, and increments $N_g$ in every iteration. At each step of the search, the algorithm solves a $\{0, 1\}$-integer program, which is a linear program for a fixed $N_g$. The search for the number of required slots stops when a schedule with the required QoS properties is found.

At each step, the algorithm solves the following $\{0, 1\}$-integer linear program:

$$
\begin{aligned}
&\text{Find} \quad \boldsymbol{\omega}, \boldsymbol{p} && && (7a)\\
&\text{s.t.} \quad \omega_j - \omega_i + \boldsymbol{\theta}_l^T \boldsymbol{p} N_f \le N_{\max} - D_l, && \forall \mathcal{P}_l \in P, \overrightarrow{f}_p(\mathcal{P}_l) = (e_i, e_j) && (7b)\\
&d_i \le \omega_j - \omega_i + p_k N_g \le N_g - d_j, && \forall c_k \in C : \overrightarrow{f}_c(c_k) = (e_i, e_j) && (7c)\\
&\quad \boldsymbol{\omega} \in \mathbb{Z}^m, \boldsymbol{p} \in \{0, 1\}^r. && && (7d)
\end{aligned}
$$

The linear program finds a feasible $\boldsymbol{\omega}$ and a feasible $\boldsymbol{p}$, ensuring a feasible schedule exists for a given $N_g$ in the iteration. The first $2q$ constraints, (7b), ensure that the total delay on all paths is less then $D_{\max}$. The next $r$ constraints, (7c), ensure that there is a feasible schedule satisfying the delay constraints. The algorithm either runs until a feasible set of $\boldsymbol{\omega}$ and $\boldsymbol{p}$ is found or until $N_g$ reaches $N_f - N_c$. Since we perform a linear search of all possible values of $N_g$, we are guaranteed to find the minimum $N_g$ for which there is a feasible schedule with a TDMA delay less than $D_{\max}$ on every path.

In order to allow the mesh routers to schedule links without the knowledge of a specific feasible $\boldsymbol{\omega}$, we introduce a new variable into the optimization. The new variable represents the maximum allowed difference between the activation time of the last link on a path and the first link on the path. We substitute $t$ instead of the first two terms in (6), so delay on the path becomes:

$$D(\mathcal{P}_l) = t + \boldsymbol{\theta}_l^T \boldsymbol{p} N_f + D_l, \quad \forall \mathcal{P}_l \in P. \tag{8}$$

The required constraint on TDMA delay, (7b), is still true if:

$$\omega_j - \omega_i \le t, \quad \forall \mathcal{P}_l \in P, \overrightarrow{f}_p(\mathcal{P}_l) = (e_i, e_j). \tag{9}$$

This leads us to the following $\{0,1\}$-integer program, to be run for each $N_g$, instead of (7):

$$\text{Find} \quad \boldsymbol{\omega}, \boldsymbol{p}, t \tag{10a}$$

$$\text{s.t.} \quad t + \boldsymbol{\theta}_l^T \boldsymbol{p} N_f \leq N_{\max} - D_l, \qquad \forall \mathcal{P}_l \in P \tag{10b}$$

$$d_i \leq \omega_j - \omega_i + p_k N_g \leq N_g - d_j, \qquad \forall c_k \in C : \overrightarrow{f}_c(c_k) = (e_i, e_j) \tag{10c}$$

$$\omega_j - \omega_i \leq t, \qquad \forall \mathcal{P}_l \in P, \overrightarrow{f}_p(\mathcal{P}_l) = (e_i, e_j) \tag{10d}$$

$$\boldsymbol{\omega} \in \mathbb{Z}^m, \boldsymbol{p} \in \{0,1\}^r, t \in \mathbb{R}, \tag{10e}$$

where the combination of (10b) and (10d) replaces (7b).

Using the symmetry between the paths we can see that (10d) is equivalent to half as many double sided constraints:

$$-t \leq \omega_j - \omega_i \leq t, \quad \forall \mathcal{P}_l, l = 1, \ldots q, \overrightarrow{f}_p(\mathcal{P}_l) = (e_i, e_j). \tag{11}$$

So when $\boldsymbol{p}$ and $t$ are fixed a feasible schedule can be found using the Bellman-Ford algorithm on a modified conflict graph. We create a new scheduling graph, $G_S(E, C_S, \overrightarrow{f}_S)$, from the conflict graph by adding arcs between the start link and the end link of the first for every path originating at the POP. This adds $q$ additional arcs to the conflict graph to create $C_S = C \cap \{c_{r+1}, \ldots, c_{r+q}\}$ arcs for the scheduling graph. The function connecting the arcs of the scheduling graph to the links $\overrightarrow{f}_S$ by combining $\overrightarrow{f}_c$ and $\overrightarrow{f}_{\mathcal{P}}$:

$$\forall c_l \in C_s, \quad \overrightarrow{f}_s(c_l) = \begin{cases} \overrightarrow{f}_c(c_l), & \text{if } l \leq r \\ \overrightarrow{f}_{\mathcal{P}}(\mathcal{P}_l), & \text{if } r < l \leq q + r. \end{cases} \tag{12}$$

Since the scheduling also has a set of inequalities associated with every arc, the schedules can be found from the scheduling graph the same way they are found from the conflict graph [6, 8].

## 6 Numerical Results

In this section, we present numerical results for the application of VoIP traffic in 802.16 mesh networks. In 802.16 mesh networks, $N_g$ is specified as the network parameter `MSH-CSCH-DATA-FRACTION` [4, p. 86]. This parameter specifies the percentage of each frame that should be used for centralized TDMA scheduling. Here, we find the percentage of the frame that should be scheduled with the 802.16 centralized scheduling protocol, so that VoIP QoS is met. The results from this section can also be used to decide the frame sizes for 802.16 mesh networks.

We assume that WTs are using the G.729 codec to encode voice. With the G.729 codec, the bandwidth of each VoIP call is 8.0kbps [7], so we assume that the end-to-end bandwidth required by each VoIP call is 8.0kbps. We use the delay budgeting presented in [7] to derive the bound on TDMA propagation delay required in the network. The delay budgeting assumes that the voice quality requires an end-to-end delay of 150ms. The delay components, not associated to voice processing, consist of the jitter buffer delay of 60ms and the Public Switched Telephone Network (PSTN) of 30ms. We assume that the PSTN delay is fixed and examine how much jitter delay can be allowed in the Internet. We use the values of $D_{\max} = 40$ms and $D_{\max} = 60$ms, corresponding to the jitter buffer delay of 20ms and 0ms, respectively.

We have generated 100 random mesh network topologies, and performed mesh network planning for each of them. Each topology was generated by placing the POP in the center of a square area of 500m$\times$500m and then randomly placing 29 mesh nodes in the square area. The topology graph for the network is created from the transmit power of the nodes and signal path loss. Each mesh node is given transmit power of 40dbm. We use the sample calculation given in [4] and the ECC-33 path loss model for medium city environments [10] to calculate the path loss due to the distance between the nodes. The modulation on each link is chosen based on received signal strength, as specified in [4, p. 765]. We assume that the network is using OFDM with 10Mhz bandwidth, so the OFDM symbol size is 25$\mu$s [4, p. 812].

The area where the mesh is located is partitioned into 25 cells, each with the radius of 50m. The purpose of the cells is to simulate short range 802.11 APs, which allow WTs to connect to the network. Each cell is assigned the mesh router closest to it as the AP. We use the minimum spanning tree algorithm to find a tree topology connecting all the mesh routers to the POP. Each router is assigned an end-to-end bandwidth to support a certain number of VoIP calls, and the end-to-end bandwidths are used to calculate link bandwidths required on every link in the network. The number of guaranteed service slots required on every link is calculated from the modulation used on the link and the symbol size.

**Table 1.** Percentage of Slots Required for VoIP Traffic ($D_{\max} = 40$ms)

| | 802.16 Frame Size | | | |
|---|---|---|---|---|
| Calls | 2.5ms | 5.0ms | 10.0ms | 20.0ms |
| 4 | 47% | 29% | 27% | 27% |
| 8 | 55% | 53% | 52% | 50% |

Table 1 summarizes the results of our numerical simulations for $D_{\max} =$ 40ms. We have used the GNU Linear Programming Kit (GLPK) [11] to perform the main $\{0, 1\}$-integer optimization in the search problem. The table represents the percentage of the slots required for VoIP traffic for 4 and 8 VoIP calls and different frame size. As observed, it is advantageous to increase the frame size since it decreases the percentage of slots needed to carry VoIP traffic. The results for $D_{\max} = 60$ms are within 2% of the values reported in Table 1. The number of slots required for guaranteed traffic does not decrease if the delay is allowed to increase up to 60ms.

## 7 Conclusion

We have presented a method to minimize the number of TDMA slots required to support a given end-to-end QoS in mesh networks. Our optimization works by performing a linear search over the number of slots required to support the given end-to-end bandwidth. At each iteration of the search, the optimization solves a $\{0, 1\}$-integer program that finds an order of transmissions in the frame, so that the maximum TDMA propagation delay is kept bellow a given QoS level and end-to-end bandwidths can be scheduled. It is important to limit the TDMA propagation delay for traffic streams such as VoIP calls, requiring a guaranteed end-to-end delay. The optimization method in this paper is appropriate for mesh network planning, since the order of transmissions can later be distributed to the nodes to create schedules. The schedules will have the same maximum TDMA propagation delay, since the delay depends on transmission ordering in the frame. We have also used numerical simulations to show the efficiency of 802.16 network in carrying VoIP traffic.

## References


1. Camp, J., Robinson, J., Steger, C., Knightly, E.: Measurement driven deployment of a two-tier urban mesh access network. Technical Report TREE0505, Rice University (2005)
2. Nortel Networks: Wireless mesh network - extending the reach of wireless LAN, securely and cost-effectively. http://www.nortelnetworks.com/solutions/wlan/ (2003)
3. Xu, S., Saadawi, T.: Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks. **39**(6) (2001) 130–137
4. IEEE: Standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems (2004)
5. IEEE: 802.11 TGs MAC enhacement proposal. Protocol Proposal IEEE 802.11-05/0575r3, IEEE (2005)
6. Djukic, P., Valaee, S.: Link scheduling for minimum delay in spatial re-use TDMA. In: Proceedings of INFOCOM. (2007)
7. Goode, B.: Voice over internet protocol VoIP. Proceedings of the IEEE **90**(9) (2002) 1495–1517
8. Djukic, P., Valaee, S.: Distributed link scheduling for TDMA mesh networks. In: Proceedings of ICC. (2007)
9. Rockafellar, R.T.: Network Flows and Monotropic Optimization. John Wiley & Sons (1984)
10. Electronic Communication Committee (ECC) within the European Conferenceof Postal and Telecommunications Administration (CEPT): The analysis of the coexistance of FWA cells in the 3.4-3.8 GHz band. ECC Report 33 (2003)
11. Makhorin, A.: GNU linear programming kit. Technical Report Version 4.8 (2005)