Scheduling Prioritized Services in Multihop OFDMA Networks

Ashima Gupta, *Member, IEEE,* Debalina Ghosh *Member, IEEE,* and Prasant Mohapatra , *Fellow, IEEE*

Abstract—Growing popularity of high speed wireless broadband access for real-time applications makes it increasingly relevant to study the admission control and scheduling of flows in a service differentiated manner. Next-generation wireless broadband networks employ Orthogonal Frequency Division Multiple Access (OFDMA) technology that enables multiple users to communicate at the same time using a time-frequency grid. In this article, we provide a mathematical model for prioritized admission control and scheduling in OFDMA based multi-hop wireless networks. The problem is formulated as an Integer Linear Program (ILP) that does joint admission control and scheduling of flows while satisfying their rate and latency requirements. We propose different heuristic algorithms for scheduling priority based flows in centralized multi-hop OFDMA networks. We define the "Flow Admittance" (FA) metric and compare the different scheduling schemes based on this metric. Simulation results show that the Start From Frame Beginning (SFB) heuristic perform well in most of the scenarios. We also propose a combination approach that merges multiple heuristics. The FA values obtained from the combination approach are close to the ILP while incurring computation time orders of magnitude lower.

Index Terms—OFDMA, Scheduling, Flow Admittance, WiMAX, Multi-hop wireless networks.

1 Introduction

A growing demand for cost effective high-speed broadband networks that are ubiquitously available has put OFDMA (Orthogonal Frequency Division Multiple Access) based wireless broadband technologies like WiMAX in the fore front of the technology drive. OFDMA offers many advantages like high data rates and reduced multipath interference over traditional radio technologies.

Convergence of the Internet with OFDMA based wireless technology to provide a multi-hop back-haul network has numerous applications like streaming audio and video. Such real-time applications demand a guaranteed quality of service(QoS) in terms of minimum rate and latency requirements to be functional. The QoS requirements of various applications are categorized into multiple service classes. These classes reflect the flexibility to rate/latency adjustment that an application can tolerate.

- *Ashima Gupta, Debalina Ghosh and Prasant Mohapatra are with the Department of Computer Science, University of California at Davis, Davis, CA, 95616 USA e-mail: ashgupta,debghosh,pmohapatra@ucdavis.edu.*

Defining a scheduling algorithm in OFDMA networks has a unique set of challenges. In OFDMA, time is synchronized and divided into timeslots. The frequency over which signals are transmitted is divided into subchannels. Bandwidth assignments can therefore be in two dimensions.

A simplistic reduction of OFDMA scheduling to a one-dimensional TDMA scheduling problem by rolling out the slots affects the latency requirements of the flows. In OFDMA-based wireless networks, interfering nodes can be scheduled on different subchannels in the same timeslot. However, half-duplex nodes cannot send and receive in the same timeslot. This distinction between interfering and half-duplex nodes is absent in TDMA where both of these kind of nodes have to use different timeslots. Hence the number of subchannels in a timeslot affect the scheduling of interfering nodes and half-duplex nodes.

Traditional metrics like throughput and delay may not depict the real impact of the scheduling schemes in prioritized service environments. For example, a scheduling scheme may admit more number of lower priority flows that have a lower minimum rate requirement and achieve high throughput. This is not beneficial to the service provider whose revenue is generated based on the Service Level Agreements (SLAs) with the customers. Additionally, service differentiation helps in provisioning quality of services for media oriented and real time applications.

We use a weighted throughput, viz. Flow Admittance (FA) metric to evaluate different scheduling schemes. The FA metric is the proportion of the weighted measure of all admitted flows to the weighted measure of all flows seeking admission. The weights assigned are linear in proportion to the priority of the service classes. This is based on the price comparison figures for different plans by various network providers. While the plans do not correspond directly to the specifications of the service classes, download speeds are part of the plans that also include different levels of security and storage. Download speeds reflect what kind of applications can be supported with the plan and typically the cost of the plan that enables real time is linear in proportion to the cost of the plan that only enables e-mail [1], [2].

The problem that we address in this paper can thus be stated: *How to admit and schedule flows for a specified network topology and number of flows with different weights, latency and minimum rate requirements, such that the "Flow Admittance"(FA) metric is maximized?* We present a mathematical model of the joint admission control and scheduling problem that satisfies the various constraints of the given network topology and flows while enabling varying link capacities. These constraints also include rate, latency, interference, half-duplex and link ordering limitations apart from the general flow conservation and capacity restrictions in network flow problems. We present an Integer Linear Program (ILP) that maximizes the "Flow Admittance". The admission control and scheduling problem is a complex version of the multi-commodity flow problem. This computationally intensive problem can therfore not be optimally solved in real-time. Hence, we propose various heuristic algorithms that admit and schedule flows. The different heuristic algorithms are:

1) Start from Frame Beginning (SFB)
2) Hopwise
3) Even-odd (E-O)

Depending on the channel conditions, subchannels on different links may use different modulation and coding rates. Thus link capacities may not be equal for all the links and usually vary with time. Our ILP and the heuristics enable varying link capacities resulting from adaptive modulation and coding rates. The combination heuristic merges all these different heuristics and outputs the best schedule. Detailed simulations are performed considering a variety of workloads and topologies. Simulation results show that FA values obtained from the combination heuristic are close to the ones obtained from the ILP while the computation time of the combination heuristic is orders of magnitude lower than the ILP. The SFB heuristic performs closest to the optimal in most scenarios.

The related works which address both bandwidth and latency in OFDMA networks are [3], [4] and [5]. However all of them are based on the even-odd framework that labels each alternate node as even and odd. Even nodes transmit in even timeslots and odd nodes transmit in odd timeslots. Narlikar, Wilfong and Zhang [4] showed that the even-odd framework can be used with wireline schedulers to guarantee delay bounds. The even-odd framework activates links in alternate timeslots. This essentially reduces the system capacity by half. Our proposed SFB heuristic does not have this limitation and is spectrally very efficient in comparison as verified by the simulations. Authors in [5] reduce the problem of meeting the latency requirements by proportionally increasing the required rate. However, this scheme penalizes other flows and results in reduced FA since many applications like VoIP do not have a high bandwidth requirement but a stringent latency requirement. Our model and heuristics differentiate between bandwidth and latency as two different parameters that results in better FA values. To the best of our knowledge this is the first work that distinguishes between and mathematically formulates per flow latency and rate requirements separately for admission control and scheduling in OFDMA networks. To summarize, the key contributions of this paper are:

1) A mathematical model of the joint admission control and scheduling problem for QoS restricted flows in OFDMA based multi-hop wireless networks. The model considers link ordering, interference, half-duplex, flow conservation and capacity constraints over multiple hops with varying link capacities. It distinguishes between latency and bandwidth as two different dimensions.
2) Presentation of the FA metric and the justification of its utility as a scheduling efficiency metric for QoS constrained flows.
3) Heuristic approaches and their comparison with the optimal and the existing even-odd algorithm. The SFB heuristic provides results close to the optimal and is realistically implementable with magnitudes lower computation time.

The rest of the paper is organized as follows. We define the network model in Section 2. In Section 3 we formulate the scheduling and admission control problem. Section 4 describes the heuristics. We present our simulation results in Section 5. In Section 6 we discuss related work. Section 7 concludes the paper.

2 Network Model

The network is modeled as a graph (V,E) where the nodes belong to the vertex set V and the links between the nodes form the edge set E. All edges are considered to be directional. Hence $e1 =< u, v >$ and $e2 =< v, u >$ are considered two different edges in our model. The terms link and edge are used interchangeably in this article.

Given a node $v \in V$, we use $E_{in}(v)$ to denote the set of edges that terminate at v.

$$E_{in}(v) = \{(w, v)|(w, v) \in E\}.$$

Similarly, $E_{out}(v)$ denotes the outgoing edges of a node v.

$$E_{out}(v) = \{(v, w)|(v, w) \in E\}.$$

For each edge e, the set $I(e) \subset E$ denotes the edges that it interferes with. $HD(e) \subset E$ denotes the edges that are in the half-duplex set of e. If $e =< u, v >$ then all edges $\acute{e}$ of the form $< w, u >$ and $< v, w >$ where $w \in V$ belong to $HD(e)$.

Each node $u \in V$ has an aggregated demand $\alpha(u)$ from its associated users. This is typical in wireless broadband networks like IEEE 802.16 where the Internet traffic is directed at the Base Station (BS). The scheduling is centralized and the nodes are static as they form a backbone network. However, the end users may be mobile.

As we are considering OFDMA networks, we assume that the frequency band is divided into K subchannels and time is slotted with a period of T. The capacity of a slot corresponding to a link e is $C(e)$. Thus the capacity of a link is $C(e) * K * T$. The traffic (flow) on a link is denoted by $D(e)$.

Let F be the set of all flows and let F_i denote the i^{th} flow. A flow is characterized by a source s_i, destination R, minimum rate requirement r_i, latency requirement l_i and weight w_i. The weight of a flow is assigned based on the service class of the flow. We use $I(e \in F_i)$ to indicate whether F_i uses edge e, that is, whether e is included in the route of F_i. Once a flow is admitted and scheduled, it follows the same schedule till its departure.

We assume that:

- The route for each flow is predetermined and non-adaptive.
- The flows are non-splittable, that is, they follow only one path. We assume this as an implication of the fixed routing and to guarantee latency constraints.
- The nodes are half-duplex, that is, they can only transmit or receive at a time.
- A specified interference estimation method that provides us with the list of interfering links. This is because providing an interference model is not the goal of this paper.
- A centralized scheduling model. Many OFDMA based wireless broadband networks like IEEE 802.16 support centralized scheduling. This is primarily because given the large range of the Base and Subscriber stations the number of hops is generally small(2 to 3).

We want to emphasize that the OFDMA scheduling problem is not the same as a TDMA scheduling problem. For example, if we want to reduce an OFDMA based system with K subchannels and t timeslots to a TDMA system, then the TDMA system will have Kt timeslots. However, not all timeslots can be treated the same as two edges in a half-duplex set cannot be assigned slots in the same time in the OFDMA system. This cannot be reflected in the TDMA system. Similarly, we have to make relevant modifications to the latency requirements of flows so that the their original QoS demands in the OFDMA system are preserved.

3 Problem Formulation

3.1 Flow Admittance

The Flow Admittance (FA) metric is defined as

$$FlowAdmittance = \frac{\sum_{i \in A} w_i}{\sum_{j \in F} w_j} \quad (1)$$

where A is the set of admitted flows, F is the set of all the flows seeking admission and w_i is the weight assigned to flow i. The numerator of the FA metric is the sum of the weights of all the admitted flows and the denominator is the sum of the weights of all the flows.

The FA metric measures how efficiently an algorithm schedules flows. An algorithm that schedules higher priority flows is better than one that schedules lower priority flows even if the number of scheduled flows is same.

3.2 Start Time Slot

Given a number of flows that need to be scheduled and their routes, we compute the start time slots of all links and all flows in the following manner. The start time slot of an edge corresponding to a flow is the timeslot within a frame by which the edge must be scheduled so that the flow reaches the destination by its deadline. Assuming each edge has the same capacity, the start time slot of a link can be defined as:

$$st^{fe} = \frac{d - fs - mt_1}{t},$$

where st is the start time slot of an edge e corresponding to a flow f, d is the deadline that is calculated based on the latency requirement of the flow, fs is the start of the next frame, m is the number of hops to the destination, t_1 includes the propagation delay and the node processing delay and t is the period of each time slot.

In the case of flows with no latency constraints, and flows whose start time slot for a link is greater than the end of the frame, st^{fe} is the last time slot of the frame. But the hop count is still taken into consideration. Thus the st^{fe} for the last link in the route is the end time slot, whereas the st^{fe} for the second last link is the end time slot - (the processing delay associated with a node and the delay associated with traversing a link).

Thus if a flow f has bandwidth requirement r and start time slots $st^{fe_1}, st^{fe_2}, st^{fe_3}$, then the flow has to be allocated $r/C(e_1)$ slots in link e_1 by st^{fe_1}, $r/C(e_2)$ slots in link e_2 by st^{fe_2} and so on.

Note that we consider that all flows that are admitted are scheduled in one frame and the scheduling repeats in all the frames in a particular scheduling period. The length of a scheduling period depends upon the overhead of disseminating the schedule information, changing link quality, queue length and other such factors. Scheduling period is an integral number of frames with the minimum being one frame. The scheduling period will vary based on the stability of the link conditions, rate increase/decrease of the subchannels and change in bandwidth demand of the traffic. Ideally, the scheduling period should be more than the cost required to compute and disseminate the schedule in the network and yet be small enough so that the current network demands are reflected in the schedule in a reasonable amount of time. Computing an optimal scheduling period and adapting it is beyond the scope of this paper and will be addressed in our future work.

Many flows will have latency greater than the frame size in a practical scenario. Hence we could have considered a superframe which consists of multiple frames and done the scheduling in a superframe. The superframe size could have been chosen based on the minimum latency required by the flows or some other appropriate criteria. For simplicity we do not consider queuing at intermediate nodes.

3.3 Statement of the Problem

The problem that we address in this paper is how to admit and schedule class differentiated flows with minimum rate and latency requirements such that the FA metric is maximized for any given topology.

We approach this problem in two steps. In the first part we assume that the flows have no latency constraints and all links have equal link capacities. We formulate an ILP to solve the same. We then extend our model to include the latency requirements and varying link conditions. The different variables used in this section are shown in Table 1.

TABLE 1
List of Notations

K	Number of subchannels
T	Number of timeslots
$C(e)$	Capacity of a slot
$D(e)$	Traffic on an edge in T timeslots
$I(e \in F_i)$	Indicates whether flow i uses edge e
A_i	Indicates whether flow i is admitted or not
Z_e	Indicates whether edge e is active in the frame
r_i	Smallest integer not less than the rate requested by flow i
$\alpha(v)$	traffic generated at node v
$I(e)$	set of interfering edges of edge e
$HD(e)$	set of half-duplex edges of edge e
X^{ets}	Indicates whether edge e is active in slot (t,s)
Y^{et}	Indicates whether edge e is active in timeslot t
$slots^{i,e}$	Number of slots assigned to flow i on edge e
X^{etsi}	Indicates whether flow i is scheduled on edge e in slot(t,s)
st^{ei}	Start time slot of flow i on edge e
$sumslots^{iet}$	Sum of slots allocated to flow i on edge e from time 0 to t

3.3.1 Formulation of the scheduling problem for flows with no latency requirements

Capacity constraint - The edge traffic $D(e)$ is the sum total of the rates of the admitted flows that use e.

$$D(e) = \sum_i r_i I(e \in F_i) A_i \ \forall e \tag{2}$$

Note that all links, e in the network may not be in the route of the considered flow. The function $I(e \in F_i)$ results in 1 if considered link e is in the route of the specified flow and 0 otherwise. We assume that the routes for each flow are predetermined based on the input topology and a routing mechanism. These routes are input to the function $I(e \in F_i)$. $D(e)$ is less than the capacity of the link in a frame.

$$D(e) \leq C(e) * K * T * Z_e \ \forall e \tag{3}$$

Note that a link may not use all K subchannels in a particular timeslot. Similarly, the edge might not be active in all the timeslots. For example, an edge e might be active for k_1 subchannels in timeslot t_1 and k_2 subchannels in timeslot t_2. Hence the total numbers of slots used by edge e is $k_1 + k_2$ and $D(e) = C(e) * (k_1 + k_2)$ which satisfies constraint 3.

Flow conservation constraint - For any node $u \in V$ that is not the destination R, the difference of the outgoing and the incoming traffic must equal the traffic generated at the node.

$$D(e)_{e \in E_{out}(v)} - D(e)_{e \in E_{in}(v)} = \alpha(v) \forall v \tag{4}$$

$$\alpha(v) = \sum_{s(i)=v} A_i r_i \forall v \tag{5}$$

$$\alpha(v) \geq 0 \forall v \tag{6}$$

Interference Constraints - Two interfering edges cannot be active in the same slot (timeslot-subchannel).

$$X^{ets} + X^{\acute{e}ts}_{\acute{e} \in I(e)} \leq 1 \forall e, t, s \tag{7}$$

Half-Duplex Constraints - The nodes in the network are half-duplex and thus they cannot transmit and receive at the same time. The half-duplex constraint is:

$$Y^{et} + Y^{\acute{e}t}_{\acute{e} \in HD(e)} \leq 1 \forall e, t \tag{8}$$

It is important to understand the difference between the interference constraint and the half duplex constraint. The interference constraint states that two interfering edges cannot be active in the same slot; however, they can use different subchannels in the same timeslot. The half duplex constraint states that if an edge e is active in timeslot t, any edge $\acute{e}$ that belongs to the set $HD(e)$ cannot be active in the same timeslot even if they use different subchannels.

Rate constraints - The minimum rate has to be guaranteed for all the edges in the path of the flow. The rate constraints can be represented as:

$$slots^{i,e} * C(e) \geq r_i * I(e \in F_i) * A_i \forall e, i \tag{9}$$

$$\sum_i slots^{i,e} = \sum_t \sum_s X^{ets} \forall e \tag{10}$$

Thus the least number of slots required to guarantee the minimum rate required by a flow should be allocated to all the edges that are in the path of that flow.

The relation between X, Y and Z are as follows:

$$Y^{et} = X^{ets_1} \cup X^{ets_2} \ldots \cup X^{ets_K}$$

$$Z^e = Y^{et_1} \cup Y^{et_2} \ldots \cup Y^{et_T}$$

Equation 1 can be reformulated as

$$FA = \frac{\sum_i w_i * A_i}{\sum_i w_i} \tag{11}$$

where A_i indicates whether flow i is admitted or not.

The ILP for flows with only bandwidth constraints is shown below:

$$maximize \frac{\sum_i w_i * A_i}{\sum_i w_i}$$

subject to the following constraints

$$D(e) = \sum_i r_i I(e \in F_i) A_i \;\forall e$$
$$D(e) \leq C(e) * K * T * Z_e \;\forall e$$
$$D(e)_{e \in E_{out}(v)} - D(e)_{e \in E_{in}(v)} = \alpha(v) \;\forall v$$
$$\alpha(v) = \sum_{s(i)=v} A_i r_i \;\forall v$$
$$\alpha(v) \geq 0 \;\forall v$$
$$X^{ets} + X^{\acute{e}ts}_{\acute{e} \in I(e)} \leq 1 \;\forall s, t, e$$
$$Y^{et} + Y^{\acute{e}t}_{\acute{e} \in HD(e)} \leq 1 \;\forall t, e$$
$$slots^{i,e} * C(e) \geq r_i * I(e \in F_i) * A_i \;\forall e, i$$
$$\sum_i slots^{i,e} = \sum_t \sum_s X^{ets} \;\forall e$$

The output of the problem is the flows that are selected and the schedule of those flows. The schedule shows the timeslots and subchannels in which a link is active. We want to emphasize that we consider OFDMA scheduling and hence a link may not use all the subchannels in a timeslot and a link may not be active in all the timeslots.

3.3.2 Formulation of Scheduling Problem for Flows with latency constraints

The capacity, flow conservation and half duplex constraints are the same as the case without latency constraints as discussed in section 3.3.1. The other constraints are given below:

Interference Constraints - Two interfering edges cannot be active in the same timeslot-subchannel.

$$\sum_i X^{etsi} + \sum_i X^{\acute{e}tsi}_{\acute{e} \in I(e)} \leq 1 \forall s, t, e \tag{12}$$

Rate Constraints - The minimum rate has to be met in all edges in the path of the flow.

$$\sum_t \sum_s X^{etsi} * C(e) \geq r_i * I(e \in F_i) * A_i \forall e, i \tag{13}$$

Latency constraints - We define the "start time slot" for all edges e in the path of a flow f. The start time slot corresponding to an edge e and flow f is the latest time slot by which the flow has to be scheduled in order to reach its destination by its deadline. Thus the latency constraint can be represented as:

$$X^{etsi} * t \leq st^{ei} * A_i \forall t, s, e, i \tag{14}$$

Link Ordering constraints - To meet the latency requirements of the flows, we need to ensure that the edges are scheduled in order. Suppose the edges from the source to the destination of a flow are labeled $e_1, e_2, \ldots e_n$. Considering a particular timeslot t, the amount of data transferred in link e_2 for flow i till timeslot t should be less than or equal to the amount of data transferred in link e_1. We use $sumslots^{iet}$ to keep the count of how many slots have been allocated to flow i on edge e upto timeslot t. Hence $sumslots^{iet} = \sum_0^t \sum_s X^{etsi}$. Also, let $d(e)$ denote the distance of the edge e from the source. Thus the link ordering constraint can be stated as:

$$sumslots^{iet} * C(e) \leq sumslots^{iut} * C(u) \forall t, e, i \tag{15}$$

where $d(e) \geq d(u)$ and $I(e \in F_i) = 1, I(u \in F_i) = 1$.

In a practical scenario, the channel conditions are not the same for all the links in a wireless network. The channel conditions also vary with time. We assume that at the beginning of each scheduling period the centralized scheduler estimates the channel condition based on the signal quality. The centralized scheduler assigns modulation and coding rates to different subchannels at different links based on channel quality. This implies that the link capacities are different in different links and hence the number of slots needed is different for different links. This is taken care of by the slot capacity $C(e)$ in our ILP.

The relation between X, Y and Z are as follows:

$$Y^{et} = \sum_i X^{ets_1 i} \cup \sum_i X^{ets_2 i} \ldots \cup \sum_i X^{ets_K i}$$
$$Z^e = Y^{et_1} \cup Y^{et_2} \ldots \cup Y^{et_T}$$

The integer linear program for flows with rate and latency constraints is shown below:

$$maximize \frac{\sum_i w_i * A_i}{\sum_i w_i}$$

subject to the following constraints

$$D(e) = \sum_i r_i I(e \in F_i) A_i \;\forall e$$
$$D(e) \leq C(e) * K * T * Z_e \;\forall e$$
$$D(e)_{e \in E_{out}(v)} - D(e)_{e \in E_{in}(v)} = \alpha(v) \;\forall v$$
$$\alpha(v) = \sum_{s(i)=v} A_i r_i \;\forall v$$
$$\alpha(v) \geq 0 \;\forall v$$
$$\sum_i X^{etsi} + \sum_i X^{\acute{e}tsi}_{\acute{e} \in I(e)} \leq 1 \forall s, t, e$$
$$Y^{et} + Y^{\acute{e}t}_{\acute{e} \in HD(e)} \leq 1 \;\forall t, e$$
$$\sum_t \sum_s X^{etsi} * C(e) \geq r_i * I(e \in F_i) * A_i \forall e, i$$
$$X^{etsi} * t \leq st^{ei} * A_i \forall t, s, e, i$$
$$sumslots^{iet} * C(e) \leq sumslots^{iut} * C(u) \forall i, e, t$$

4 Heuristic Scheduling Algorithms

In this section we describe three heuristics which differ in the way they assign the slots to flows. The heuristics are:

- Start from Frame Beginning (SFB)
- Hopwise
- Even-odd (E-0)

Each heuristic schedules flows for a scheduling period. A scheduling period consists of an integral number of frames, the minimum being one frame. The same schedule is followed in all the frames of a scheduling period. We calculate the minimum number of slots required by a flow in a frame based on its rate requirement and the capacity of a slot. The flows are scheduled according to the service class priority. Within each service class flows are scheduled by arrival time.

All the algorithms try to maximize the FA value in most cases by using a priority based greedy strategy. Let us consider flows of three different service classes (1,2 and 3) with weights w_1, w_2 and w_3 where $w_1 > w_2 > w_3$. Let there be N_1 number of flows with weight w_1, N_2 number of flows with weight w_2, and N_3 number of flows with weight w_3. For simplicity let us assume that the slot requirements of all the flows are the same.

Let us assume that n_1 number of flows of service class 1, n_2 number of flows of service class 2 and n_3 number of flows of service class 3 are accepted. Also, $n_1+n_2+n_3 = N$. Hence the Flow Admittance is

$$FA1 = \frac{w_1 n_1 + w_2 n_2 + w_3 n_3}{w_1 N_1 + w_2 N_2 + w_3 N_3} \quad (16)$$

This is essentially the same as the flow admittance definition in equation 1. We denote the denominator by D. Now when the algorithms admit the $N + 1$th flow, they choose a service class 1 flow. Hence the Flow Admittance becomes

$$FA2 = \frac{w_1(n_1 + 1) + w_2 n_2 + w_3 n_3}{D} \quad (17)$$

Suppose the scheduling was done in a different manner and a flow of service class 3 was selected. The FA metric will be

$$FA3 = \frac{w_1 n_1 + w_2 n_2 + w_3(n_3 + 1)}{D} \quad (18)$$

As $w1 > w2 > w3$, $FA2 > FA3$. If no more flows can be scheduled, then choosing the highest priority flow maximizes the FA value.

However the heuristics do not always maximize the FA metric. Let $2w_2 > w_1$. Suppose instead of choosing the $N + 1$th flow of service class 1, they choose a flow of service class 2 and suppose the rate requirement of this flow is much less than the flow of service class 1. This enables another service class 2 flow to be scheduled. Hence

$$FA4 = \frac{w_1 n_1 + w_2(n_2 + 2) + w_3 n_3}{D} \quad (19)$$

Thus $FA4 > FA2$. However as the algorithms always pick flows by priority, they will not schedule flow of service class 2 instead of service class 1. Hence in this case the heuristics do not maximize the FA metric.

The Even-Odd algorithm adopts the Even-Odd framework proposed by [4] where every node in the routing tree is alternately labeled as even or odd. Considering only uplink traffic and directional edges we label the edges even or odd based on the hop count from the base station. However, this essentially halves the system capacity as even edges are only scheduled in even timeslots and odd edges are scheduled in odd timeslots.

The SFB heuristic removes this restriction so that any link can be scheduled in any timeslot provided that the half-duplex constraints (equation 8) is satisfied and latency requirements of flows are met. The flows with latency requirements have to reach their destination within the required deadline. Hence for each flow and for each link in the path of the flow, there is a start time slot as discussed in section 3.2. Thus an entire flow is scheduled from the source to the destination before the next flow is considered. The timeslots allocated to a flow is less than or equal to the start time slot for the flow and edge ensuring the latency constraint (equation 14) is met. The SFB algorithm assigns timeslots starting from the frame beginning and continues assigning slots till the required slots are assigned or the start time slot is exceeded. We also considered an alternative approach where we started allocating slots from the start time slot and proceeded towards the beginning of the frame. However the simulation results show that the differences between these two approaches are negligible and hence we have omitted the discussion of the second approach here.

Both the E-O and the SFB heuristics schedules all the links in the path of a flow before considering the next flow. The Hopwise algorithm adopts a different approach where we try to simultaneously schedule all the flows of a particular class.

The algorithms try to allocate the minimum number of slots requested by the flow to all the links in the path of the flow to ensure constraint 9. As a flow is admitted and the schedule is made permanent only if the flow can be allocated the minimum rate from the source to the destination, the flow conservation constraint is also maintained. Also the bandwidth allocated to flows is based on the capacity of the assigned slots and this preserves the capacity constraints. Flows are scheduled in an interference-aware manner. Our heuristics can work with any given interference model. Thus, two interfering links are not scheduled in the same timeslot-subchannel. When a slot s is selected for allocation, it is checked whether any edge in the interfering set is allocated the same slot.This is to ensure that the interference constraint is not violated. Similarly, when a slot is selected for allocation, it is checked whether any edge in the half-duplex set uses the same timeslot. This ensures that the half-duplex constraint is preserved. If an edge in the half-duplex set uses the timeslot, then a different timeslot is considered. Link ordering is preserved in all the heuristics even though the capacities of the links may be different. This is done in the following manner. Suppose the route for a flow f from source to destination consists of two edges e_1 and e_2. Let the capacity of a slot for e_1 be c_1 and the capacity of a slot for e_2 be c_2. At time t if e_1 has been allocated s_1 slots, then at time $t + 1$, e_2 can be allocated $\frac{s_1 c_1}{c_2}$ slots. Thus the heuristics maintain the number of slots allocated to any edge of a flow till a particular timeslot and uses that variable to compute the number of slots that can be allocated to the next edge.

4.1 Start from Frame Beginning (SFB) Algorithm

The SFB Algorithm considers flows one by one based on service class priority and arrival time. It tries to allocate the minimum number of slots required to all the links in the path of a flow. The slots allocated are stored in the temporary data structure, Tempschedule. If the number of slots allocated is equal to the number of slots required, the flow is accepted and Tempschedule is made permanent, otherwise the flow is rejected. Algorithm 1 shows the ScheduleFlow function used in the SFB algorithm. The AllocateSlots function iterates over all

```
for each flow f in F do
    slotsAllocated = AllocateSlots(f);
    if slotsAllocated == slots required then
        Make Temporary Schedule Permanent;
        accept flow f;
    else
        reject flow f;
    end
end
```

Algorithm 1: ScheduleFlow (SFB)

the links of the flow and assigns slots based on the interference, half-duplex and link ordering constraints. The number of slots required is computed by taking into account the link capacity which can change per link depending on the channel conditions. AllocateSlots starts searching for free slots for each link of the flow from the beginning of the frame and proceeds towards the start time slot which is the last time slot by which a flow must be scheduled in a link so that it meets its deadline. It also considers the number of slots allocated to the previous edge in the route of the flow to ensure that the link ordering constraints are met. If there are no free slots in a particular timeslot, then the next timeslot is considered.

SFB preserves the link ordering constraints as well as flow conservation constraints. Hence, the total amount of data transmitted in an edge e1 by any timeslot for a specific flow is always less than or equal to the total data transmitted in edge e2 by that timeslot for the same flow; given that e1 is closer to the source than e2.

4.2 Hopwise

The Hopwise heuristic considers all the flows in a particular class and assigns slots to the links associated with the flow hop by hop. Algorithm 2 shows the ScheduleFlow Function for the Hopwise heuristic.

The ScheduleFlow function takes all the flows in a particular service class C and attempts to assigns slots for all these flows for a particular hop h in the routing tree. This iterative process ends when all the hops in the routing tree have been considered. Thereafter, the flows whose minimum bandwidth and latency requirements have been met for all the links in their path are accepted

```
for each flow f in a service class C do
    h = MAX-HOP;
    while h >=0  slotsAllocated(f,h) =
    AllocateSlotsHopwise(f,h);
    h=h-1;
end
for each flow f in a service class C do
    if   for each link l in f  do
        h = hopCount(l);
        slotsAllocated(f,h) == slots required;
    end
    then
        Make Temporary Schedule Permanent;
        accept flow f;
    else
        reject flow f;
    end
end
```

Algorithm 2: ScheduleFlow (Hopwise)

and the schedule is made permanent. Then this process is repeated for the next service class. The AllocateSlotsHopwise(f,h) function is similar to AllocateSlots function used in the SFB algorithm but it uses the hopcount h as the parameter. Thus AllocateSlotsHopwise function allocates slots for only one edge of a flow at a time depending on the hop count of the edge.

4.3 Even-Odd Algorithm

The Even-Odd Algorithm labels alternate nodes as even and odd. Thus the directional links between nodes may be labelled even or odd depending on whether it is originating from an even or odd node. Even links are activated in even timeslots and odd links are activated in odd timeslots. This ensures that a node does not transmit and receive in the same timeslot. However, this essentially halves the capacity of a link. The E-O algorithm is based on the even-odd framework proposed by [4]; however, our implementation of this algorithm schedules flows closest to the deadline, that is, starts allocating slots from the start time slot and proceeds towards the beginning of the frame. The E-O algorithm also considers the flows one by one based on their service class priority and arrival time. Once a flow is considered for allocation, the links through which the flow will traverse is considered. If the link under consideration is labeled even, the algorithm starts looking at even time slots beginning from the start time slot of the link and the frame. Alternately, if the link is odd, only odd timeslots is considered. Once the required minimum number of slots have been allocated to all the links of the flow, the flow is admitted. This iterative process continues for all the flows.

4.4 Combination Approach

None of the heuristics will give the best result for every traffic scenario and for every topology. Hence, one approach can be to combine multiple heuristics and get the schedule corresponding to the heuristic that gives the best FA values. The decision which heuristics to combine may be based on which ones have the best FA values for the observed common traffic scenarios.

5 SIMULATION RESULTS

IEEE 802.16 based WiMAX technology uses the OFDMA physical layer. Therefore, for simulation purposes we utilize the IEEE 802.16 framework. The IEEE 802.16 standard defines five different service classes of traffic as illustrated in Table 2. We assume the same in our simulations.

TABLE 2
Service Classes in WiMAX

Class	Application	QoS parameters
Unsolicited Grant Service (UGS)	VoIP,E1; fixed-size packets on periodic basis	max rate, latency and jitter
Real-Time Polling Service (rtPS)	Streaming audio/video	minrate, maxrate and latency
Enhanced Real-Time Polling Service (ertPS)	VoIP with activity detection	minrate,maxrate, latency and jitter
Non Real-Time Polling Service (nrtPS)	FTP	minrate and maxrate
Best Effort (BE)	Data transfer, Web	maxrate

A WiMAX network consists of a central entity called the Base Station (BS) and several nodes called the Subscriber Stations (SS) or Relay Stations (RS). In a multihop WiMAX network SSs communicate with the BS using multiple hops [6], [7]. The network architecture in a Mobile Multihop Relay (MMR) network is a tree with BS as the root. We assume a similar network in our evaluations. The number of nodes in such networks is usually small because of the large range of the BS, SSs and RSs.

We use ILOG CPLEX 10.0 for modeling and solving the mathematical formulation of the problem. Our model and simulations can work with any user provided topology and interference model. However, for illustration purposes we assume two topologies, viz. a balanced binary tree (Figure 1), and an unbalanced tree (Figure 2) and the protocol interference model which includes primary and secondary interference. Note that the model, heuristics and simulation environment are independent of the topology. While we have used two topologies to generate(illustrate) the results, the topology and interference matrix are user inputs. The FA value depends on the constraints, topology and the flow set characteristics. In general the FA value generated by a heuristic for a specific flow set and topology would depend primarily on the interference matrix. The link ordering and half duplex constraints are topology dependent. Hence for a given topology and flow set, the heuristics result should depict the same trend as illustrated by the two topologies considered, that SFB would perform close to the optimal as the most common case. However we can always use the combination heuristic which always provides the best result among all the heuristics as the final output.

We implement the heuristic algorithm using a custom simulator written in C. We can perform both dynamic and static flow scheduling. For purposes of comparison with the optimal solution we used static scheduling to ensure that the input to both algorithms is identical. Flows are generated according to Poisson arrival process. The lifetime of a flow is exponentially distributed and used to generate the departure time of the flows. The number of flows generated per set could be different each time because the type of flows are also generated randomly. Each type is associated with a different bandwidth and latency range. We use a uniform random generator to generate the source of the flow. In these simulations, we consider only uplink flows. Both programs can be easily extended to schedule for both uplink and downlink by randomly choosing a destination, obtaining the route based on the topology and adding more timeslots for the downlink subframe. The frame length and maximum number of subchannels are configurable.

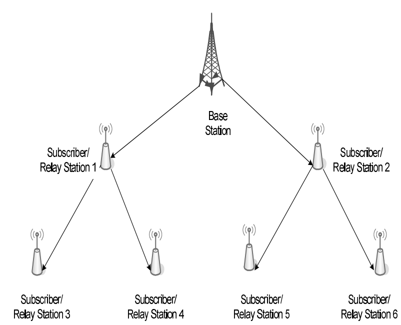


Fig. 1. Balanced Binary Tree Topology.

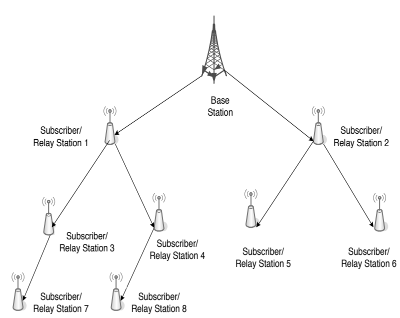


Fig. 2. Unbalanced Tree Topology.

In all the heuristic algorithms, we implement five global queues at the BS ordered by priority (UGS highest and BE lowest). Within each queue, flows are ordered on the basis of arrival times. Each flow has an associated route that is computed from the source and destination generated by the flow generator and the topology input by the user. The schedule for each link is computed by the algorithm and stored as a two dimensional structure of subchannels indexed by timeslots. Flows are picked for scheduling in service class priority and within each service class by first arrival time. In all the algorithms the goal is to allocate only for the minimum bandwidth requirements of the flows. Unused capacity can be distributed among the flows in service class priority manner in all the cases.

We generated different sets of flows for which the Optimal and all the heuristics were executed. The flow mix in each set was varied from 10% of one service class to 100% of that service class and the rest divided equally among all the other service classes. For data sets that required over eighteen hours of computation time by the ILP, we omitted the results. For obtaining results illustrating the effect of varying link rates resulting from adaptive modulation we generated multiple rate sets. Each rate set contains the rate for each link in the network. The rates would be assigned by the BS but for the simulation we chose the rate in a uniform random manner from the set of rates available for WiMAX uplink. We compare the results based on the *Flow Admittance* metric. We ran our simulations on an Intel Pentium 2.2 GHz Linux machine with 1 GB memory.

5.1 Flow Admittance as a Comparison metric

We use Flow Admittance as the metric of comparison for scheduling instead of throughput. Flow Admittance is a weighted throughput and captures class differentiation. We illustrate this by the following example. We refer to the ILP that maximizes the FA metric as Optimal-FA or simply Optimal and the ILP that maximizes the throughput as Optimal-Throughput. The Optimal-Throughput problem has identical constraints as the Optimal-FA ILP except that we maximize the throughput.

$$maximize \sum_i r_i * A_i$$

where r_i is the rate requirement of flow i.

In our simulation, for example, priority of UGS > ertPS > rtPS > nrtPS > BE. On the other hand, typically the rate requirement of nrtPS > rtPS > ertPS > UGS > BE. We obtain the results for both metrics in terms of the number of accepted flows for each service class for some representative flow sets with latency requirements for the balanced binary tree topology.

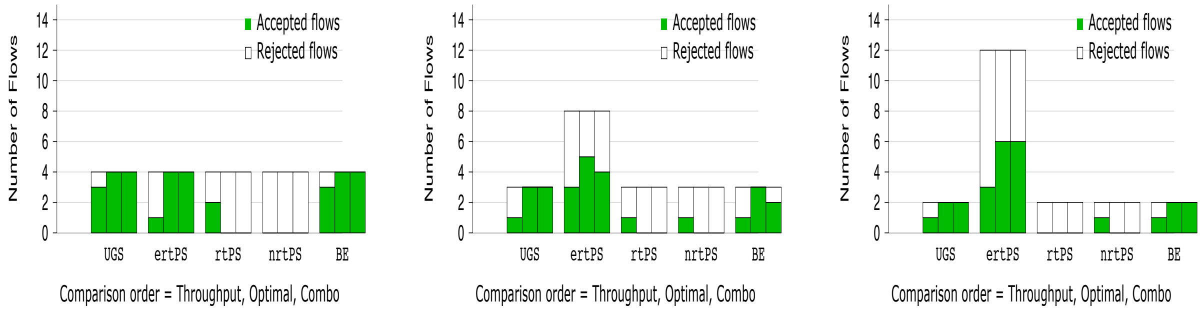
As we can see from Figure 3, the ILP that maximizes FA metric is a better measure for determining scheduling efficiency of class differentiated flows. On the other hand, the throughput maximizer ILP is better at scheduling nrtPS flows that have a higher rate requirement in the generated flow sets. If the bandwidth requirement of the flows was in proportion to their service class priority, then the results would be similar for both the ILPs, since FA is essentially weighted throughput.

5.2 Flow Admittance Results

5.2.1 Identitical link capacities

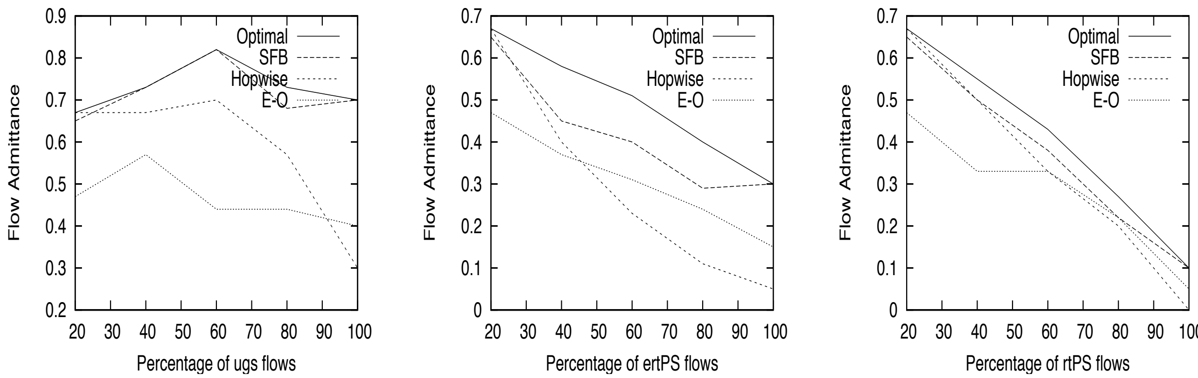
In this section we provide the comparison results of the heuristics with the ILP that includes latency constraints. While we obtained results of the ILP and the heuristics with bandwidth only constraints, they did not provide any additional insight hence in the interest of space we did not include them. The number of slots used is 64, with 8 timeslots and 8 subchannels. A higher number of slots results in magnitudes higher convergence time for the ILP while the computation time for the heuristics remains linear. For example, with 32 subchannels and 10 timeslots the CPLEX program took a few hours to generate the results for just one set of flows. In contrast, the heuristics took milliseconds to generate the results with as many as 256 subchannels and 20 timeslots. All the links in the topology were assigned the same lowest link capacities, which was the same for all links. We obtained the values for the FA metric using the ILP that includes the latency constraints. We compared these values to the FA values obtained from the heuristics.

Figures 4 and 5 characterize the FA metric with varying percentage of UGS, ertPS, rtPS and nrtPS flows in the flow sets. We observe that in figure 4 the FA value increases up to the 60% composition of the UGS flows for all the algorithms except the E-O. This increasing trend is because of acceptance of larger number of higher priority UGS flows in the flow sets. UGS flows have only moderate bandwidth requirements, hence bandwidth availability is not a bottleneck. Beyond the 60% mark the FA values of the algorithms decreases since the latency requirements of the larger number of UGS flows cannot be met. However the FA value for the E-O decreases from the 40% mark as it has lower capacity available to schedule. In case of the SFB algorithm, the FA value rises a little after the 80% mark. This is because it accepts one more flow in the 100% case that has a source only one hop away from the BS as compared to the 80% case in which all the remaining flows were two hops away and hence their latency requirement could not be met in the available number of slots. *We would like to point out that the purpose here is to do a relative comparison of the FA values obtained by the heuristics for any given scenario with the optimal. The curve followed by the FA values for a specific algorithm in this case may change depending on the assigned weights, link capacities, topologies, interference and other factors that are part of the scenario.* For the other service classes, we observe a decreasing trend for the FA values as the percentage of a particular service class increases in the flow mix. This decrease is caused by the reduction in resources required by the flows of that particular service class. We observe that FA values for the E-O algorithm is better than the SFB and Hopwise



(a) Flow set I (b) Flow set II (c) Flow set III

Fig. 3. Balanced Tree: Flow acceptance comparison (bandwidth and latency).



(a) Varying percentage of UGS flows (b) Varying percentage of ertPS flows (c) Varying percentage of rtPS flows

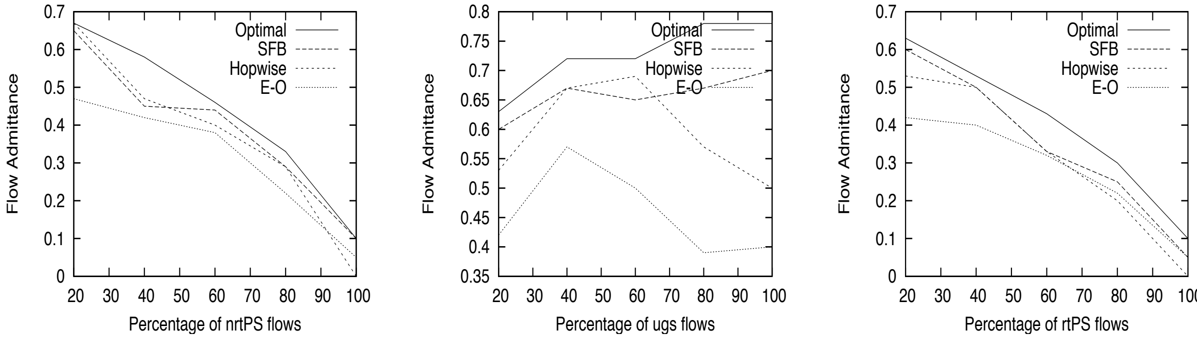
Fig. 4. Balanced Tree: FA characterization for various flow sets (bandwidth and latency).

algorithms when the percentage of rtPs flows increase in the rtPs curve. This is because as the rtPs flows have very stringent latency requirements, the latency becomes the bottleneck and the decreased capacity of the E-O algorithm does not have any effect. On the other hand, as E-O algorithm schedules flows from the start time slot and proceeds towards the beginning of the frame, it performs better than the SFB when the flow mix contains mostly rtPs flows with very stringent latency requirements.

We also observe the Hopwise Algorithm performs very poorly when the percentage of rtPs flows increase in the flow mix. The Hopwise algorithm schedules slots for all the flows in a service class for one hop, before attempting to schedule the other hops for those flows. This results in the poor FA values when the majority of the flows belong to the same service class, since there may be very few slots left for scheduling the next hop. This will result in most of the flows being rejected as their requirements for the entire route will not be met. Figures 5(b) and 5(c) illustrate the results for varying percentage of UGS and rtPS flows in the unbalanced tree topology. Results for varying percentage of UGS show an initial increase in the FA values for all algorithms, followed by a little divergence in behavior at the 40% mark. The SFB heuristic show an increase after the 40% mark as it schedules in a per flow manner irrespective of the hops and in the scenario is left with more capacity to schedule. The Hopwise schedule flows per hop and hence is unable to find slots to meet the latency requirements of the UGS flows as the percentage of these flows increase beyond 70% in the flow mix. The E-O too shows a decrease after the 40% mark as it is limited by the capacity it can schedule. The Optimal is always higher. As can be seen for increasing percentage of rtPS flows the FA value decreases. Both SFB and E-O converge at the 0.05 FA value when the flow set comprises only of rtPS flows since at this point latency becomes the limiting factor rather than the bandwidth. The Hopwise heuristic results in rejection of all flows as it tries to schedule one hop for all the flows and exhausts all suitable slots for a single hop so that no suitable slot is left for other hops. We also obtained the results for unbalanced tree topology for varying percentage of ertPS and nrtPS flows. The results indicated similar trends.

5.2.2 Varying link rates

In this section we illustrate the effect of varying link rates resulting from adaptive modulation and coding rates on scheduling of various flow sets in the balanced tree topology. We assume that the BS will assign the modulation and coding rates based on the estimated signal quality values after a scheduling period. The scheduling period may be varied based on the overhead

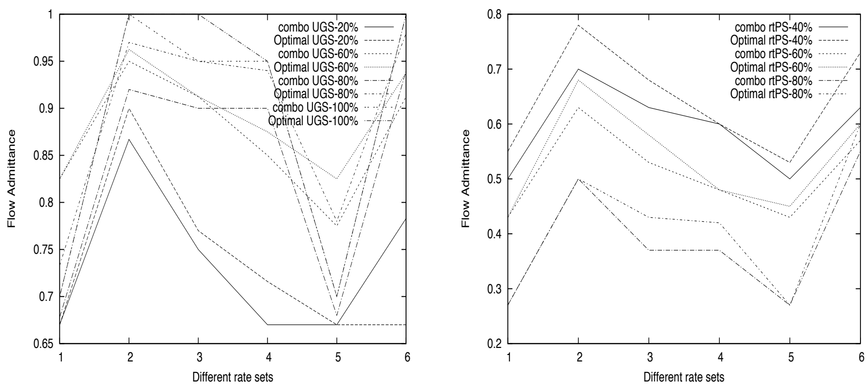


(a) Balanced Tree: Varying percentage of nrtPS flows

(b) Unbalanced Tree: Varying percentage of UGS flows

(c) Unbalanced Tree:Varying percentage of rtPS flows

Fig. 5. FA characterization for various flow sets (bandwidth and latency).



(a) Varying percentage of UGS flows

(b) Varying percentage of rtPS flows

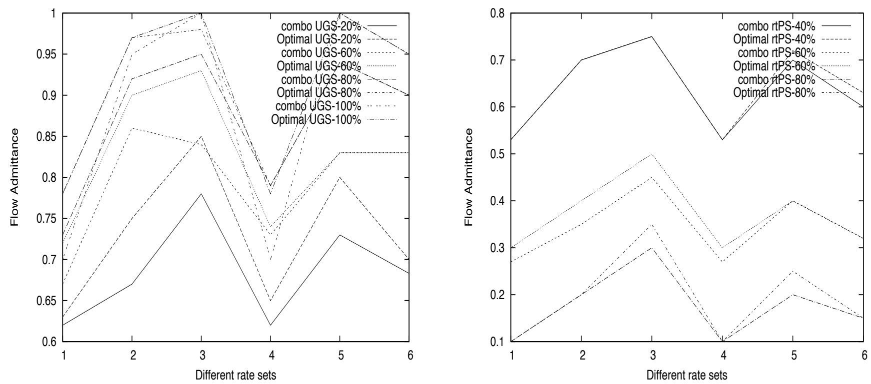
Fig. 6. Balanced Tree: FA characterization with multiple rate sets (bandwidth and latency).

and error rate trade offs. We refer to the set of rates chosen for all the links of the topology as a rate set. In this section we consider the effect of different rates on multiple flow sets in the topology over a scheduling period. The rate for a link is randomly chosen from the set of possible rates for WiMAX uplink for a specific frequency as shown in Table 3. Note, that this is not the same as frequency selectivity where a better subchannel will be chosen over others for scheduling a flow based on criteria like CINR/RSSI values, bandwidth traffic etc. That is orthogonal to our work.

TABLE 3
Uplink Data Rate at Various Modulation and Code Rate

Modulation and Code Rate	5 MHz	10 MHz
QPSK, 1/2	653	1,344
QPSK, 3/4	979	2,016
16 QAM, 1/2	1,306	2,688
16 QAM, 3/4	1,958	4,032
64 QAM, 1/2	1,958	4,032
64 QAM, 2/3	2,611	5,376
64 QAM, 3/4	2,938	6,048
64 QAM, 5/6	3,264	6,720

We generated five rate sets and obtained FA values using all the heuristics and the ILP for each of these rate sets with varying percentage of UGS and rtPS flow sets. We considered a number of flow sets for each class (UGS and rtPS). The results are depicted in Figure 6 and 7. We note that in Figure 6(a), generally with increasing percentage of UGS flows in the flow set, the FA value typically increases for a specific rate set, except in rate set 5. The reason for this increase has been explained in section 5.2.1. All the links in rate set 1 have the same rates (the lowest rate). The bottleneck links (the links connecting the BS to other nodes) in rate set 5 have the same capacities as rate set 1. Hence, rate set 5 is almost equivalent to rate set 1 and the capacity available for scheduling is much less in rate set 5 as compared to the other rate sets. Therefore, the requirements of all the UGS flows are not met. The FA values for rate set 5 is very similar to the FA values of rate set 1. For a particular flow set, however the FA values depend on the number of flows passing through a link, the link rate as well as the requirements of the flows. For example, on comparison of rate set 5 with rate set 6 we determined that the rate for the bottleneck link between SS1 (Subscriber station 1 in Figure 1) and the BS are the same in both set and hence the number of flows originating from the left side of the tree are constrained by that. However, the rate for the link between SS2 and the BS as well between SS6 and SS2 quadruples in rate set 6 as compared to rate set



(a) Varying percentage of UGS flows (b) Varying percentage of rtPS flows

Fig. 7. Unbalanced Tree: FA characterization with multiple rate sets (bandwidth and latency.)

5. Hence the FA values are much higher with rate set 6 than rate set 5 for all flow sets by both the combination and Optimal algorithms. Similarly in Figure 6(b), as the percentage of rtPS flows increases in the flow set the FA values for both Optimal and the combination heuristic drops as explained earlier. Results for the unbalanced topology depicted in Figure 7 show the same trend, even though the two results for balanced and unbalanced tree cannot be directly compared since the rate sets were generated randomly and are different for both.

5.3 Number of Accepted Flows

In this section we compare the type of flows accepted by the different algorithms for two representative flow sets in the Balanced tree topology. Every flow set comprises of equal number of UGS, ertPS, rtPS and nrtPS flows. Figure 8 depicts the accepted flows of all service classes in two flow sets obtained for the different heuristics. It is clear from the figures that the SFB algorithm is better at scheduling flows of higher priority classes than the Hopwise and E-O algorithms. For example, the SFB algorithm schedules more rtPs flows than Hopwise and E-O in flow set I. Similarly, it can schedule some nrtPs flows whereas Hopwise is unable to schedule any nrtPs flows.

5.4 Computation Time

Computation time is the time taken by the scheduling algorithm to execute. We generated flow sets with varying number of total flows while keeping the flow mix uniformly distributed. We then executed each of the scheduling algorithms on every flow set. The results are depicted in Figure 9. The computation time for all the heuristic algorithms increase linearly with the number of flows. Note that the computation time for the Optimal is not shown in the figure as the Optimal takes hours on average whereas the computation time required by the other algorithm is in the order of milliseconds. We ran our simulations on a 2 GHz Intel Core 2 Duo processor and 1 GB 667 MHz DDR2 SDRAM memory. Note that

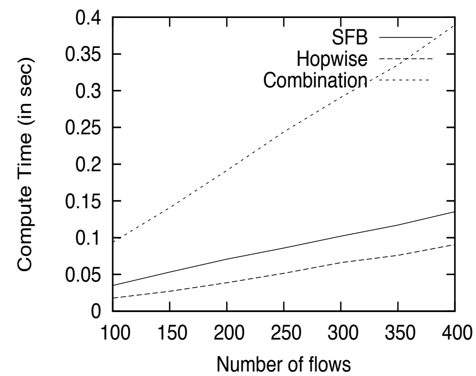


Fig. 9. Computation Time of the Heuristics.

we increased the number of subchannels to 64 and the number of timeslots to 16 to accomodate the larger number of flows.

5.5 Observations and Inferences

From our evaluations we observed that the FA metric is more useful than the throughput metric to evaluate the efficiency in scheduling prioritized flows. The combination heuristic chooses the schedule that gives the best FA value among all the heuristics. The SFB heuristic also performs well in most scenarios and hence we can use SFB heuristic to obtain a good result in most cases. While FA values for the E-O heuristic follow the same trend, they are significantly lower than the other algorithms since the even-odd framework halves the network capacity. The computational time of all the heuristic algorithms is order of magnitudes less than that of the Optimal. We also observed that as the number of flows of a particular service class increases in a flow set, the FA values decreases because of diminished resource availability for that class. This decrease is sharper as flows with higher bandwidth requirements increase in the flow set, for example, nrtPS flows and rtPS flows. However, as the number of UGS flows increases in a flow set, the FA value initially increases since UGS flows have the highest priority and usually the lowest bandwidth

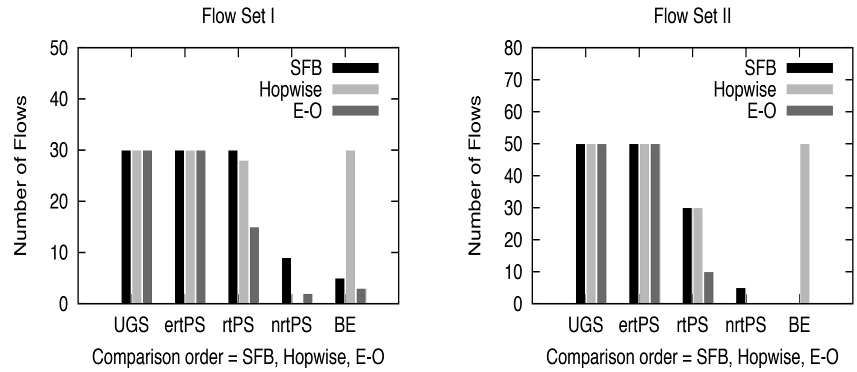


Fig. 8. Characterization of Flow Acceptance by the different heuristics.

requirements. From our simulations we determined that there is a high impact on the FA values for different link data rates.

6 Related Work

Some of the scheduling schemes for wireless multi-hop networks schedule only one link at a time and hence they do not effectively utilize the capacity of the network [8], [9]. Many other works have focused on TDMA link scheduling and they do not take sub-channelization into account ([10], [11], [12], [13]). Distributed scheduling in multihop wireless networks has been addressed in [14], [15], [16] and [17]. The objective of [18] is to formulate a scheduling problem which maximizes the system throughput under the fairness model defined by the authors. Jin et al. [19] address the problem of routing and packet scheduling for throughput maximization in IEEE 802.16 mesh networks. Tang, Xue and Zhang([20]) studied bandwidth allocation in multi-channel multihop wireless mesh networks. They try to maximize network throughput and enhance fairness at the same time. Scheduling and resource allocation for an OFDMA-based wireless network is addressed in [21]. A centralized scheduling scheme using multiple channels and single transceivers in a WiMAX Mesh Network is discussed in [22]. The goal is to minimize the length of scheduling defined as the number of timeslots needed to complete all the data transmissions. The authors in [23] propose a packet scheduling scheme in WiMAX Mesh Networks using bidirectional concurrent transmissions. Kwak and Cioffi in [24] focus on the subchannel allocation problem with power constraints for maximizing the sum-rate in downlink multi-hop OFDMA relay networks. They model the optimal power allocation problem for fixed subchannel subsets by a modified water-filling algorithm. Li and Liu model the optimal source/relay/subcarrier assignment problem that maximizes the sum rate from all sources to the destination, with fairness constraint for OFDMA relay based networks in [25]. They do not consider the QoS requirements and prioritization of flows. Our model does not consider power allocation. We model the OFDMA channel as a subchannel-timeslot grid so that it is not necessary to use a two slot alternate relay transmission assignment wherein source nodes transmit in the first slot and relay nodes transmit in the second slot. Our model includes QoS characteristics and class-differentiation of flows. We also include spectral reuse. In our model subchannel-timeslot slots can be reused for non-interfering links. Papers [26] [27] [28] focus on single hop downlink systems. We have modeled multi-hop uplink networks. A simple and generalized even-odd framework for link activation is proposed in [4] where the authors present techniques for constructing interference-free routes within the scheduling even-odd framework. The scheme requires that a route does not contain two interfering even or odd links. Thus, there may not be a feasible route in their scheme even if two nodes are able to communicate with each other. The authors in [5] present an admission control scheme for flows with rate and latency constraints. They use the even-odd framework to guarantee the latency constraints and use dynamic programming to find the admitted flows. They associate rewards with each flows and try to maximize the total reward. However, they do not give the schedule of the flows. A heuristic based admission control and scheduling scheme based on the even odd framework is presented in [3]. However, the scheme does not do link ordering and hence some flows may not be able to meet their latency requirements.

7 Conclusion

In this paper we formulated an analytical model of the joint admission control and scheduling problem for QoS constrained flows in OFDMA based multi-hop wireless networks. The model

- Distinguishes between bandwidth and latency constraints of flows as two different dimensions.
- Accounts for link ordering constraints.
- Considers interference, half duplex constraints, flow conservation and capacity constraints.
- Addresses all the above constraints over multiple hops with varying link capacities.

We presented the "Flow Admittance" metric, which provides a proportion of the weighted measure of all admitted flows to the weighted measure of all flows seeking admission. The weights are assigned depending on the priority of the service class of the flows. Our simulation results demonstrated the effectiveness of using this metric over a traditional throughput only metric for measuring the schedule efficiency in a prioritized scheme.

We proposed several heuristics and compared their results with the Optimal as obtained from the model. The heuristics are efficient in terms of computational time and scalability. We also proposed a combination approach that merges multiple heuristics and outputs the best schedule. Our heuristics can be easily extended to include any fairness policy.

REFERENCES

[1] ISP, Website, 2008, "http://isp1.us/dial-up/".

[2] AT&T, Website, 2008, "http://www.att.com/gen/".

[3] D. Ghosh, A. Gupta, and P. Mohapatra, "Admission control and interference-aware scheduling in multi-hop wimax networks," in *IEEE MASS*, 2007.

[4] G. Narlikar, G. Wilfong, and L. Zhang, "Designing multihop wireless backhaul networks with delay guarantees," in *Proceedings of Infocom*, 2006.

[5] S. Lee, G. Narlikar, M. Pal, G. Wilfong, and L. Zhang, "Admission control for multihop wireless backhaul networks with qos support," in *Proceedings of WCNC*, 2006.

[6] D. Bayer, N. v. Waes, and C. Eklund, "Tutorial:802.16 mac layer mesh extensions overview," Website, 2002, "http://grouper.ieee.org/groups/802/16/tga/contrib/S80216a-02_30.pdf".

[7] R. B. Marks, "IEEE 802 tutorial: 802.16 mobile multihop relay," Website, 2006, "http://www.ieee802.org/16/sg/mmr/".

[8] A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," in *MSWiM*, 2006.

[9] H. Shetiya and V. Sharma, "Algorithms for Routing and Centralized Scheduling to Provide QoS in IEEE 802.16 Mesh Networks," in *WMuNeP*, 2005.

[10] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, "Interference-aware IEEE 802.16 Wimax mesh networks," in *Vehicular Technology Conference*, 2005.

[11] B. Han, W. Jia, and L. Lin, "Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks," *Computer Communications.*, 2007.

[12] P. Djukic and S. Valaee, "Scheduling algorithms for 802.16 mesh networks," in *Wimax/MobileFi: Advanced Research and Technology*, Y. Xiao, Ed. Auerbach Publications, CRC Press, 2007.

[13] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient interference-aware tdma link scheduling for static wireless networks," in *Proceedings of ACM Mobicom*, 2006.

[14] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu, "Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode," in *Mobihoc*, 2005.

[15] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *Proceedings of IEEE Infocom*, 2007.

[16] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," in *Proceedings of IEEE Infocom*, 2007.

[17] N. Bayer, B. Xu, V. Rakocevic, and J. Habermann, "Improving the performance of the distributed scheduler in ieee 802.16 networks," in *VTC*, 2007.

[18] M. Cao, V. Raghunathan, and P. Kumar, "A Tractable Algorithm for Fair and Efficient Uplink Scheduling of Multi-hop WiMAX Mesh Networks," in *WiMesh*, 2006.

[19] F. Jin, A. Arora, J. Hwang, and H.-A. Choi, "Routing and Packet Scheduling for Throughput Maximization in IEEE 802.16 Mesh Networks," in *IEEE Broadnets*, 2007.

[20] J. Tang, G. Xue, and W. Zhang, "Maximum throughput and fair bandwidth allocation in multi-channel wireless mesh networks," in *Proceedings of IEEE Infocom*, 2006.

[21] R. Agarwal, R. Berry, J. Huang, and V. Subramanian, "Optimal scheduling for ofdma systems," in *Proceedings of 40th Annual Asilomar Conference on Signals, Systems and Computers*, 2006.

[22] P. Du, W. Jia, L. Huang, and W. Lu, "Centralized Scheduling and Channel Assignment in Multi-Channel Single-Transceiver WiMax Mesh Network," in *WCNC*, 2007.

[23] Q. Xiong, W. Jia, and C. Wu, "Packet Scheduling Using Bidirectional Concurrent Transmission in WiMAX Mesh Networks," in *WiCOM*, 2007.

[24] G. Li and H. Liu, "Resource allocation for ofdma relay networks with fairness constraints," in *IEEE Journal on Selected Areas in Communications*, 2006.

[25] R. Kwak and J. M. Cioffi, "Resource-allocation for ofdma multi-hop relaying downlink systems," in *Proceedings of IEEE GLOBECOM*, 2007.

[26] K. L. C. Y. Wong, R. S. Cheng and R. D. Murch, "Multiuser ofdm with adaptive. subcarrier, bit, and power allocation," in *IEEE Journal on Selected Areas in Communications*, 1999.

[27] J. Jang and K. B. Lee, "Transmit power adaptation for multiuser ofdm systems," in *IEEE Journal on Selected Areas in Communications*, 2003.

[28] Y. L. M. Tao and F. Zhang, "Resource allocation for delay differentiated traffic in multiuser ofdm systems," in *IEEE Transactions on Wireless Communications*, 2008.

PLACE PHOTO HERE

Ashima Gupta is a Ph.D student at UC Davis in the Department of Computer Science. Her research focus is on Scheduling for Energy Conservation and Quality Enhancement in Multi-hop Wireless Networks, including WiMAX, Sensor and Wi-Fi networks. Prior to this she obtained her Masters in Computer Applications from University of Pune in India.

PLACE PHOTO HERE

Debalina Ghosh is a Ph.D student in the Department of Computer Science at the University of California, Davis. Prior to this she obtained her Masters in Computer Science at the University of Illinois at Chicago and her undergraduate degree in Computer Science and Engineering from Jadavpur University, India. Her research interest is in scheduling and coexistence issues in wireless networks.

PLACE PHOTO HERE

Prasant Mohapatra is currently the Tim Bucher Family Endowed Chair Professor and the Chairman of the Department of Computer Science at the University of California, Davis. In the past, he has been on the faculty at Iowa State University and Michigan State University. Dr. Mohapatra received his doctoral degree from Penn State University in 1993, and received an Outstanding Engineering Alumni Award in 2008. Dr. Mohapatra's research interests are in the areas of wireless networks, sensor networks, Internet protocols, and QoS.