Providing QoS to Real and Data Applications in WiMAX Mesh Networks

Vinod Sharma, A. Anil Kumar, S. R. Sandeep,
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore, India. 560012
Email: { vinod, anilkumar, srs@pal }@ece.iisc.ernet.in,

M. Siddhartha Sankaran
Department of Computer Science
National Institute of Technology
Tiruchirapalli, India. 62015
siddharthasankaran@gmail.com

***Abstract*— We consider the problem of centralized routing and scheduling for IEEE 802.16 mesh networks so as to provide Quality of Service (QoS) to individual real and interactive data applications. We first obtain an optimal and fair routing and scheduling policy for aggregate demands for different source-destination pairs. We then present scheduling algorithms which provide per flow QoS guarantees while utilizing the network resources efficiently. Our algorithms are also scalable: they do not require per flow processing and queueing and the computational requirements are modest. We have verified our algorithms via extensive simulations.**

Keywords: Mesh networks, WiMAX networks, end-to-end QoS, Joint routing and scheduling.

I. INTRODUCTION

IEEE 802.16 standard [1], also known as WiMAX supports a Mesh mode (the other mode being point to multipoint) in which unlike the traditional cellular systems, the nodes can communicate without having a direct connection with the base station. This improves coverage and data rates even on uneven terrain.

In a IEEE 802.16d Mesh network, a node that has a direct connection to backhaul services outside the Mesh network, is termed a Mesh Base Station (MBS). All other nodes of a Mesh network are termed Mesh Subscriber Stations (MSS). In IEEE 802.16d standards these nodes are stationary (however 802.16e supports mobility). The standard specifies a centralized scheduling scheme for mesh networks. But it does not specify algorithms for routing and scheduling, which have a significant impact on the performance of the system and will largely decide the end to end QoS to different users. In this paper we develop such algorithms.

The problem of scheduling and routing in adhoc multihop wireless networks has been extensively studied in recent years (see [3], [10], [17] for general surveys and tutorials).

The studies on multihop 802.16 networks are [7], [8], [14], [23], [24] and [26]. In [26] a simple heuristic scheduling and a Tree routing algorithm are proposed to achieve efficient channel utilization. [7] and [14] provide fair access to all nodes and also efficient utilization of resources. In [24] also routing and scheduling algorithms are provided which are efficient for the overall system but spatial reuse of the channels is not allowed (because the 802.16 standard at that time did not allow spatial reuse). In [23] also channel spatial reuse is not allowed but within this limitation the authors provide QoS to individual TCP and real time connections. The QoS guarantee to individual flows has not been provided in any other multihop wireless network study that we are aware of (all the other studies mentioned above provide scheduling and routing for the *aggregate* traffic generated at different nodes which as we will see is not sufficient to guarantee QoS to individual connections). In [8] the authors study the distributed scheduling.

In this paper, we present algorithms for centralized scheduling of real and non-real time traffic with the objective of providing QoS within the framework of the IEEE 802.16 mesh mode. We first obtain an optimal and fair routing and scheduling of the aggregate traffic generated at different nodes within the network. Then we fix the real time and TCP connections that pass through a particular link and also the slots in which the link transmits. Next we develop algorithms that each link uses to schedule the transmission of the packets of different flows passing through it on the slots assigned to it so as to provide QoS to individual flows. Our algorithms use the network resources efficiently and fairly and can be used in real time by the MBS. The main difference between this paper and [23] is that in this paper we allow channel spatial reuse which can significantly increase the system capacity.

We have also developed an admission control policy in our setup. In addition, we have tested our algorithms when some of the nodes are mobile. These details are omitted due to lack of space but are available in [6].

The organization of the paper is as follows. Section II describes the system model. We obtain an optimal and fair routing and link scheduling algorithm in Section III. In Section IV we develop scheduling algorithms to provide QoS to UDP (real time) connections. TCP connections will be studied in Section V. In Section VI we handle both UDP and TCP traffic together to provide QoS to each connection. Section VII validates our claims via simulations. Section VIII concludes the paper.

II. SYSTEM MODEL

IEEE 802.16 supports two modes of operation: Point to multipoint (PMP) and Mesh mode. In PMP the traffic is transmitted directly between the BS and an SS (Subscriber Station). In the Mesh mode, the overall geographical area is divided into meshes. Each mesh has a Mesh BS. The other nodes in a mesh are called Mesh subscriber stations (MSSs). A transmission can take place between two MSSs within a mesh or in two different meshes. The transmission between two MSSs within a mesh can occur via other MSSs within the mesh which may or may not involve the MBS. Transmission between two MSSs in two different meshes involves transmission from the source MSS to its MBS (possibly via other MSSs in the mesh), from the MBS to BS, from BS to the MBS of the receiver mesh and finally from this MBS to the receiver MSS.

In this paper we consider the mesh mode. The mesh mode supports two different physical layers, WirelessMAN-OFDM and WirelessHUMAN. Both of these use 256 point FFT OFDM TDMA/TDM for channel access and operate in a frequency band below 11GHz. The first operates in the licensed band but the second uses the unlicensed band. The standards also support adaptive modulation and coding.

The mesh mode supports only Time Division Duplex (TDD) to share the channel between the uplink and the downlink. The standards support both centralized and distributed scheduling of slots. Centralized scheduling is mainly used to transfer data between the MBS and the MSSs. Thus centralized scheduling is the dominant mode because most of the communication takes place with nodes outside the mesh. In this mode, the MBS periodically collects the channel information and the resource (throughput) requests of all the nodes to draw up the schedule which it distributes to the nodes.

We consider the following scenario. Consider a Mesh network with M MSSs labeled 1, 2, ..., M. The MBS is labeled 0. We consider Uplink and Downlink *Centralized Scheduling* of the MSSs, which, according to the standards uses TDMA with spectral reuse. Also the data is directed either to or from the MBS. We assume that each node transmits at the maximum allowed power, if needed. (Although power control is also an important issue the standard currently does not emphasis it). As the channel condition on a link changes, the data rate is also changed so as to meet the desired BER (Bit Error Rate). Let r_{ij} denote the rate and E[r_{ij}] the average rate of the channel from node i to node j. Resource allocation is done by the MBS in units of (mini) time slots. One time slot transmits multiple OFDM symbols. Each allocation is valid for K frames consisting of N time slots (for simplicity of notation we will take K=1).

To provide the QoS, we will generally follow the QoS-architecture developed in [23] since this seems to be the only architecture available for 802.16 mesh networks which guarantees per flow QoS. However, [23] did not consider spectral spatial reuse which can significantly improve capacity. Thus, in our current proposal we will remove this restriction.

We will use a two step approach. In the first step we will provide routing and scheduling for the *aggregate* traffic for each source-destination pair of MSSs (one of these MSSs will be the MBS). This of-course does not guarantee the QoS to individual flows. In the second step we develop scheduling algorithms to share the long-term throughput guaranteed in step one between real and data applications to guarantee QoS to individual flows.

Section III provides the routing and scheduling for step 1. In Sections IV-VI we develop step 2. Due to space limitations some details are not provided (see however [6], [20]).

III. ROUTING AND SCHEDULING AGGREGATE TRAFFIC

The algorithms developed in this section can be used for uplink as well as downlink simultaneously. Let $\lambda(s,d)$ be the mean number of bits per slot to be transmitted from MSS s to MSS d. This is the sum of mean throughput required by all the real time and data connections transmitting from MSS s to MSS d. We develop algorithms which will decide the routes that $\lambda(s,d)$ will follow and also the slots in which each link will transmit.

Our algorithms are functions of $\lambda(s,d)$ and the mean link rates $E[r(i,j)]$ but otherwise will not vary with time. By exploiting the current queue lengths at different links and the channel states one could vary the routes and the schedules to obtain better performance ([22], [23]) but we will not do this. (See [6], [20] for justification). These algorithms will be run at the MBS and then the schedules broadcast to different nodes via Mesh Centralized Schedule messages.

The algorithms we develop will satisfy the traffic requirements $\lambda(s,d)$ of each source-destination pair (s,d) if possible. If not, then we will provide a "fair" solution which is also efficient.

In this section we use the approach developed in [20] which in turn was partly motivated by [15] and [16].

In [23], where spatial reuse is not allowed it was shown that the routing and scheduling problems can be decoupled and that a Tree structure can be optimal for routing. In the present general scenario this may not be true (although the 802.16 standard seems to prefer the Tree structure ([7], [26])). In [6] we have also developed an algorithm which provides an optimal Tree to satisfy the aggregate requirements of different (s,d) pairs (if possible) and provides performance better than the algorithm in [26]. However, the optimal Tree performance is worse than the solution provided here.

The cost function to optimize will be a sum of the link cost functions $f(\Gamma(i,j), n(i,j))$ where $\Gamma(i,j)$ is the total mean traffic rate per slot and $n(i,j)$ is the fraction of slots assigned to link (i,j). Often f will be nonlinear. For example, using Kleinrock's independence assumption ([25]) or approximating the queues at each link by an $M/M/1$ queue, we get

$$f(\Gamma(i,j), n(i,j)) \quad = \quad \frac{\Gamma(i,j)}{n(i,j)E[r(i,j)] - \Gamma(i,j)} \qquad (1)$$

as the mean queue length at the link (i,j) and $[n(i,j)E[r(i,j)] - \Gamma(i,j)]^{-1}$ as the mean delay. Similarly we can consider packet loss probability if the buffer lengths are small. Using Lagrange multipliers one can accomodate constrained optimization (see [20] for more details and also for other cost functions).

We consider the following joint routing and scheduling problem:
Find $n(i,j)$ and $\alpha_p(s,d)$ that minimizes

$$\sum_{(i,j)\in\mathcal{L}} f(\Gamma(i,j), n(i,j)) \qquad (2)$$

subject to

$$\Gamma(i,j) = \sum_{(s,d)} \sum_{p:(i,j)\text{is on } p} \alpha_p(s,d)\lambda(s,d) \le n(i,j)E[r(i,j)], \qquad (3)$$

$$0 \le \alpha_p(s,d) \quad \text{for each } p, (s,d) \qquad (4)$$

and

$$\sum_p \alpha_p(s,d) = 1 \quad \text{for each } (s,d) \qquad (5)$$

where $\alpha_p(s,d)$ is the fraction of (s,d) traffic on route $p, \mathcal{L}$ is the set of links and the inner summation in (3) is over all possible routes for (s,d). The condition (3) is required to satisfy the stability condition at each link (i,j).

Obtaining the optimal solution in (2)-(5) can be very time consuming because of the nonlinear cost function. Also, if it is not possible to satisfy the $\lambda(s,d)$ requirements of each (s,d), the above optimization problem may not provide any solution. Thus in the following we first develop algorithms which will check for feasibility of the demands $\lambda(s,d)$. If these are not feasible, then we provide a solution which may be "fair" to all (s,d) pairs. Finally we obtain a solution which is fair to all (s,d) pairs and optimizes the nonlinear cost function.

Consider the following optimization problem:

$$max \lambda \text{ such that} \tag{6}$$

$$\sum_p \alpha_p(s,d) \geq \lambda \text{ for all } (s,d) \tag{7}$$

and (3) and (4) are satisfied where the summation in (7) is over all possible paths p in the network.

A solution to the above optimization problem can be considered "fair" and efficient. This is because if there is a routing and scheduling algorithm which satisfies all the traffic requirements $\lambda(s,d)$ then λ will be ≥ 1. If not, it provides the largest fraction of traffic that can be satisfied for each (s,d). This concept of fairness has also been considered in [16] and [23]. Furthermore unlike (2)-(5), this problem is a linear program (LP) and hence can be solved much faster than the nonlinear problem (2)-(5).

In addition to (3)-(7) the network should also satisfy some transmission constraints. These constraints occur due to wireless nature of the links. Sometimes we can write these constraints as necessary and/or sufficient linear inequality constraints. For example, if no spatial channel reuse is allowed then necessary and sufficient conditions are

$$\sum_{(i,j)} n(i,j) \leq 1. \tag{8}$$

It is shown in [20] that if a node can receive successfully if and only if one of its neighbouring nodes transmits in a slot and that a node can transmit only on one of its links at a time, then the necessary and sufficient conditions are

$$\sum_{j:(i,j)\in\mathcal{L}} n(i,j) \leq 1 \quad \forall \;\; i \;\; \text{and} \;\; \sum_{i=(i,j)\in\mathcal{L}} n(i,j) \leq 1 \quad \forall \;\; j. \tag{9}$$

If we put the constraint that only one incoming or outgoing link at a node can be active at a time, then ([7]) necessary and sufficient conditions, are

$$\sum_{j:(i,j)\in\mathcal{L}} n(i,j) + \sum_{j:(j,i)\in\mathcal{L}} n(j,i) \leq 1 \text{ for all nodes } i. \tag{10}$$

Our general setup can work with transmission constraints of the type (8)-(10). The problem of transmission constraints has also been studied in [4], [12] and [16]. (We will denote the particular constraint considered at a time as (T)).

In general the scheduling problem is NP hard because the $n(i,j)$ need to be integer valued (should be the number of slots in a frame assigned to link (i,j)). However if we ignore the integrality of $n(i,j)$ and consider them as non-negative fractions as considered above, (3), (4), (6), (7), (T) becomes an LP problem which is computationally much more tractable.

If the number of nodes in the mesh is large, then complexity of the above LP can also be of concern because the number of variables $\alpha_p(s,d)$ can be exponential in number of nodes. Then we can reformulate the problem as in [4], [2], [15].

Next we consider the minimization of the cost function

$$\sum_{(i,j)\in\mathcal{L}} f(\Gamma(i,j), n(i,j)) \tag{11}$$

while satisfying (3), (4), (T) and

$$\sum_p \alpha_p(s,d) \geq \lambda \text{ for all } (s,d) \tag{12}$$

where λ is the optimal solution obtained from LP (3), (4), (6), (7) and (T). This is a nonlinear optimization problem and can be quite computationally intensive. (See [20] for some efficient algorithms).

One can further improve the efficiency of the system if the optimal λ in (6) is less than 1. In [20] an algorithm is provided where the fraction of demands satisfied for some of the (s,d) pairs can be increased without decreasing the fraction for other (s,d) pairs below the optimal λ obtained above.

The routing and scheduling provided above will ensure that the average rate $n(i,j)E[r(i,j)]$ is sufficient to carry the overall traffic passing through link (i,j). However, it will not ensure that the throughput (rate) seen by traffic of a pair (s,d) will indeed get its required share of throughput. To ensure this, we will store the total traffic of different (s,d) pairs passing through a link in different queues at that link and provide the required throughput to each queue via WRR (Weighted Round Robin).

Based on the solution of the routing and scheduling algorithm we know the fraction $\alpha_p(s,d)$ of total average traffic requirement $\lambda(s,d)$ of each pair (s,d) passing through a route p. Then, based on the *average* throughput requirement of each UDP and TCP connection of (s,d), we will decide which of the CBR, VBR and TCP connections of (s,d) will pass through which route. Knowing this, we decide the following QoS architecture.

IV. QOS FOR REAL TIME TRAFFIC

In this section we design scheduling algorithms to guarantee QoS to individual UDP connections. Two important real time applications are IP telephony and video conferencing. For these applications, the end to end delay of a packet should not exceed (say) 150 msec. Also, the fraction of packets dropped for an application should be less than (say) 2%. To satisfy these QoS requirements, we propose that at the end of a (scheduling) frame we drop the packets which could not be transmitted through the wireless network. This will ensure a maximum delay of about 40 msec (for $K = 2$ and each frame of 10 msec) in the wireless network (the rest of the delay margin is left for the remaining part of the network that a packet may have to travel). We develop algorithms which will ensure that no user will experience drop probability greater than 2%.

Audio encoders usually generate CBR (constant bit rate) traffic while the video encoders (e.g., MPEG) generate a VBR traffic. We consider these traffics separately.

A. Scheduling of CBR traffic

Let X (a constant) be the total amount of traffic generated during a frame by different CBR connections of a particular (s, d) pair following a particular route denoted by links $p_1, p_2, ..., p_h$ (this will be known based on the algorithm in Section III). Let the upper bound required on the drop probability of the packets of these flows be ϵ.

The scheduling problem for this CBR-UDP traffic is to calculate the number of slots $n_j, j = 1, \ldots, h$ required at link p_j such that X units of data can be transmitted to the MBS per scheduling frame and the end to end drop probability is bounded by ϵ.

This problem was addressed in [23]. However, it was observed that in practice, due to very small probabilities ϵ, the number of slots needed actually becomes X/r_{min} where r_{min} is the minimum rate supported in the standard. This makes the analysis independent of the link statistics. Similar comments will hold for VBR scheduling.

B. Scheduling of VBR Traffic

Consider K VBR flows generated at an (s, d) that will follow the same route $p_1, p_2, ..., p_h$. Let $D_n(k)$ be the amount of data generated by flow k in frame n. We assume that the arrival process $\{\, D_n(k), n \geq 0\,\}$ for each $k = 1, \ldots, K$ is stationary and ergodic with known statistics and independent of each other. The problem is to calculate the number of slots required by this VBR traffic on each node on its route in order to bound the drop probability by ϵ. For this problem, in [23], first we obtain *equivalent bandwidth* ([25]) of the VBR sources and then use that to obtain the number of slots as for the CBR sources.

V. QOS FOR TCP TRAFFIC

Some applications using TCP, e.g., web traffic and file transfer may require certain minimum response time. We try to satisfy these QoS requirements by providing adequate minimum mean throughput to individual TCP connections.

We consider the case of persistent TCP connections. These are long lived connections which need to send a large file. We have also considered TCP-ON-OFF connections (see [11] for details on this model) which model the web traffic using HTTP 1.1. Due to lack of space we will not present this model (see [6], [23]).

Let N^P persistent TCP connections of an (s, d) be passing through a particular route. Let λ_j^P be the minimum throughput requirements (in packets/sec) and s_j^P the packet lengths (in bits) of the j^{th} TCP connection. Thus the total average throughput requirement of the TCP connections is $\lambda^P = \sum_{j=1}^{N^P} \lambda_j^P s_j^P$ bits/sec.

Let the N^P TCP connections are passing through (say) four queues (Fig.1). TCP_i has window size W_i (initially assume that it is fixed) and propagation delay Δ_i (representing delays in the rest of the network). At each queue the link speed is c bps (ensured say, by WRR discussed above).

It is shown in [21], [22] that the total throughput obtained by TCP_i is

$$\frac{W_i\, c}{\sum_{j=1}^{N^P}(W_j - \lambda_j\, \Delta_j)\, s_j^P + 3\, s_i^P + \Delta_i\, c} \text{ packets/sec.} \qquad (13)$$

However this may not provide the QoS to different TCP connections. For that we control the mean window size $E[W_i]$ of different connections via RED ([9]). These ideas have been presented in [19] and [23] and we briefly explain here.

Let us fix a desired queueing delay of d^* sec in the first queue. Define for each i, $\widetilde{\Delta_i} = \Delta_i + \frac{3\, s_i^P}{c}$. We fix the desired mean window size of E[W_i] such that

$$\frac{E[W_i]}{d^* + \widetilde{\Delta_i}} = \lambda_i^P \quad \text{for each } i. \qquad (14)$$

Now we use RED control for each TCP connection i and specify its RED parameters such that at average queue length $d^*\, c$, it will drop the packets of TCP_i with probability p_i, where

$$p_i = 8/(3\,(E[W_i] + 4)^2 + 5)$$

for each i (this formula has been used in [23] and is based on [18]. Better approximations are available in [5]). Then it can be shown (see [21] and [22]) that this system will have a steady state such that the first queue will have the mean queue length $d^*\, c$, the queue lengths at the other queues will be negligible and each of the TCPs will have their mean window size E[W_i] satisfying the above requirements. Furthermore, TCP_i will get the throughput λ_i^P packets/sec.We will verify these claims via simulations in Section VII.

It is shown in [13] and [19], that the TCP connections can be grouped such that one needs only a few RED parameters to take care of the throughput requirements of different TCPs and per flow processing is not required.

VI. JOINT SCHEDULING OF UDP AND TCP FLOWS

From the arguments in Sections IV and V in order to provide QoS to UDP we had to consider the worst case channel conditions whereas for TCP we had to consider the average channel rates. Thus there is a huge difference between the total average bandwidth requested and the total average bandwidth *provided* to guarantee the QoS of the CBR and VBR connections. Here we utilize this extra bandwidth for scheduling of TCP flows.

We provide priority to UDP traffic over TCP traffic in the network. It has been observed in [13], [19] and [23] that by doing this the delays experienced by UDP flows can be

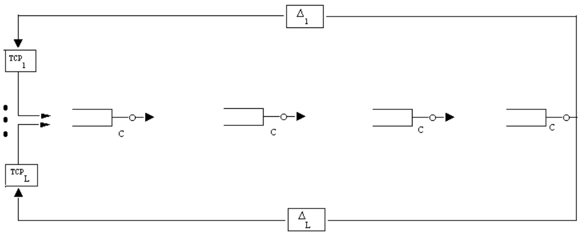


Fig. 1. Multiple TCP flows through multiple queues with fixed rates

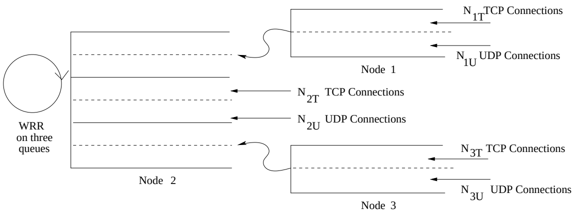


Fig. 2. Overall Scheduling of TCP and UDP flows to provide QoS. UDP flows get priority over TCP flows in each queue. RED applied on TCP flows at entry nodes. (We have separated UDP flows and TCP flows in each queue by a dotted line indicating each queue might be implemented via two queues.)

drastically reduced without affecting the throughput of the TCP flows.

Let $\lambda_U(s,d)$ and $\lambda_T(s,d)$ be the *average* throughput required by the total UDP (for a CBR connection it is its rate, for a VBR connection, it is its equivalent bandwidth) and the total TCP traffic generated by (s,d). Then we define $\lambda(s,d) = \lambda_U(s,d) + \lambda_T(s,d)$ as the average requirement of (s,d). We use this requirement in Section III to obtain the routing and scheduling for all the pairs. It is possible that the overall traffic of (s,d) is split over several routes. Let $\alpha_p(s,d)$ be the fraction of (s,d) traffic on route p.

If $\alpha_p(s,d) < 1$ then we will send $\alpha_p(s,d)$ fraction of $\lambda_U(s,d)$ and $\lambda_T(s,d)$ on route p. On a link (i,j) on p out of a total $n(i,j)$ slot assignment in Section III, a fraction $n'(i,j,s,d) \geq \frac{\lambda(s,d)\alpha_p(s,d)}{E[r(i,j)]}$ would be assigned for $\lambda(s,d)\alpha_p(s,d)$ traffic of (s,d).

Once the UDP and TCP flows of (s,d) to be routed on route p have been identified, we can compute the fraction of slots $n_U(i,j,s,d)$ needed on a link (i,j) on route p to satisfy the QoS of those CBR and VBR connections via the methods detailed in Section IV. Finally we assign $n(i,j,s,d) \overset{\triangle}{=} max(n_U(i,j,s,d), n'(i,j,s,d))$ fraction of slots of link (i,j) for the traffic of (s,d) passing through it. Also, we give priority to the real time traffic of (s,d) over the TCP traffic of (s,d) on each link (i,j). Thus, since $n(i,j,s,d) \geq n_U(i,j,s,d)$, the QoS of the real time traffic will be satisfied. Also, because $n(i,j,s,d) \geq n'(i,j,s,d)$, the long term average rate of TCP and UDP traffic of (s,d) on (i,j) will be satisfied and hence the TCP traffic gets its share of throughput on each link (i,j). Often $n(i,j,s,d)$ will be much less than $n_U(i,j,s,d) + \frac{\lambda_T(s,d)\alpha_p(s,d)}{E[r(i,j)]}$ (the number of slots needed on link (i,j) to satisfy the QoS of TCP and UDP flows of (s,d) if we do not give priority to UDP flows) and hence this multiplexing of UDP and TCP traffic of (s,d) on (i,j) provides significant gains in resource requirement. At present the traffic mix in Internet is such that the real time traffic makes less than 10% of the overall traffic. It is likely to be so in near future. In that case $n(i,j,s,d)$ will most likely be close to $n'(i,j,s,d)$ and hence the optimal solution obtained in Section III will not need to be modified.

In Fig. 2 we show our overall QoS architecture when there are three nodes. Node i has N_{iU} UDP flows and N_{iT} TCP flows entering the Mesh. Flows from nodes 1 and 3 enter node 2. At node 2 the flows from the three nodes are stored in separate queues and are provided their required throughput via WRR. In each queue the UDP packets are given priority over the TCP packets. The TCP flows are controlled via RED algorithms at their entry nodes.

Our scheme is scalable because we do not need to do per flow processing (see details in [13], [19] and [6]).

VII. SIMULATIONS

We consider the network shown in Fig. 3. There are 20 MSSs and 1 MBS (at the center). The network characteristics are summarized in Table I. The system provides adaptive modulation and coding rates resulting in 96 to 432 bytes per mini slot. The scheduling is done over 3 frames. The channel rates vary randomly from one scheduling period (of 3 frames) to another independently. The channel rates are varied with Rayleigh distribution and then mapped to one of the rates. In Fig. 3, the numbers written on the links represent the link rates. If the number is x, then the mean link rate is $48x$ bytes per mini slot. We consider the transmission constraints (9).

Figs. 4 and 5 provide the details on the number of TCP and UDP flows respectively used in the example. The first 20 columns in Fig. 4 indicate the number of TCP flows transmitting data from 20 MSSs to the MBS. The next 20 columns indicate the number in the downstream. There are 5 classes of TCP flows depending upon their throughput requirements which are 16, 32, 64, 128, and 256 Kbps respectively. Similarly the number of UDP connections are shown in Fig. 5. These are all sending traffic in the upstream.

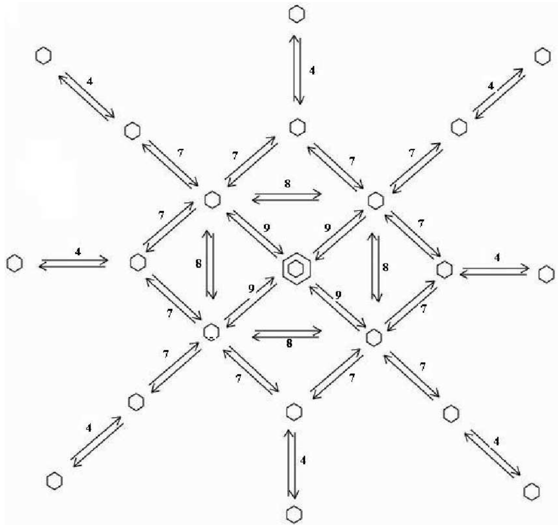
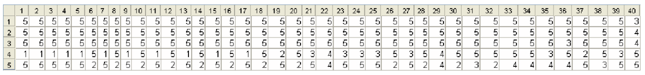


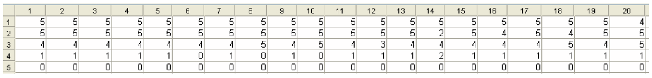
Fig. 3. Network used in simulations



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	1	1	1	1	1	5	1	5	1	5	1	5	1	5	1	5	1	5	2	5
5	5	5	5	5	5	2	5	2	5	2	5	2	5	2	5	2	5	2	5	2

	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4
3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	5	5	5	4
4	3	4	3	3	3	5	3	5	4	5	5	5	5	5	3	5	2	5	3	5
5	5	4	5	5	5	2	5	2	4	2	3	2	4	4	4	4	5	3	5	5

Fig. 4. Number of TCP connections



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4
2	5	5	5	5	5	5	5	5	5	5	5	5	5	2	5	4	5	4	5	5
3	4	4	4	4	4	4	4	5	4	5	4	3	4	4	4	4	4	5	4	5
4	1	1	1	1	1	0	1	0	1	0	1	1	1	2	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 5. Number of UDP connections

TABLE I

PHYSICAL LAYER PARAMETERS

Bandwidth	20 MHz
Number of Subcarriers	256
Frame Duration	10ms
No. of OFDM symbols / frame	844
No. of OFDM symbols / minislot	4
Total No. of minislots / frame	211
No. of minislots / frame for uplink Centr. Sched.	194

TABLE II

PERFORMANCE OF TCP FLOWS

Percent Error	Fraction of Flows
$<$ -20	0.052
-20 to -15	0.034
-15 to -10	0.041
-10 to -5	0.058
-5 to 0	0.083
0 to 5	0.1133
5 to 10	0.086
10 to 15	0.087
15 to 20	0.060
$>$ 20	0.382

There are 5 classes of UDP connections depending upon their requirements. The first 3 represent CBR traffic sending data at rates 16, 32, 64 Kbps respectively. The last 2 classes represent VBR traffic sending data at mean rates 128 and 256 Kbps respectively.

The packet sizes of the CBR connections are 100 Bytes, of the VBR connections are 1500 Bytes and of the TCP connections are 1000 Bytes. The TCP connections also have an extra propagation delay of 0.05 sec to account for the delays of the packets and the acks in the rest of the network. The acks for TCP packets are sent at higher priority. Each VBR source is Markov modulated with transition matrix

$$\begin{pmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.2 & 0.2 \\ 0.2 & 0.3 & 0.2 & 0.3 \\ 0.1 & 0.4 & 0.2 & 0.3 \end{pmatrix}$$

and with rates 73 kbps, 146 kbps, 293 kbps and 586 kbps in the four states for a source with a mean rate 256 kbps. For a source with a mean rate 128 kbps, in each state the rate is half of that for a 256 kbps source.

From the LP in Section III, we obtain that the given aggregate requirements are feasible. From $\alpha_p(s, d)$, we identified the CBR, VBR and TCP connections of (s, d) that will use the route p. Also from the $n(i, j)$ for each link (i, j), we obtain the allocation of slots to different links in each scheduling period of three frames. In the following we provide the performance of this LP solution. For this example we did not run the nonlinear optimization (11)-(12). However see [6], [20].

We observed no packet losses for CBR and VBR connections in our simultations. The average delay of the CBR packets was less than 28.01 msec and of the VBR packets was less than 28.09 msec. Table II provides the error= Achieved average throughput - minimum required throughput for the different TCP connections. From these results we see that the QoS of the real and data traffic is being met very well. This has happened when most of the links are heavily loaded.

We have simulated several other networks with different traffic mixes and have made similar observations, (see [6], [20]).

VIII. CONCLUSIONS AND EXTENSIONS

In this paper we have designed efficient, fair and practically implementable algorithms for routing and centralized scheduling in IEEE 802.16 mesh networks. We provide end to end QoS to different flows in the network. For this, we first provide an optimal and fair joint routing and scheduling solution to satisfy the *aggregate* mean traffic requirements of different source-destination pairs. Then we do scheduling at individual links to provide QoS to each flow. For this, we have handled UDP and TCP traffic separately at first and then jointly.

Our algorithms are able to provide QoS to real and nonreal time individual flows efficiently and fairly. We are also able to support limited mobility as envisaged in the 802.16e standard. We have also provided an admission control policy which is an important part of any QoS framework. These details are provided in [6].

REFERENCES

[1] Air Interface for Fixed broadband Wireless Access Systems, *IEEE STD 802.16*, October 2004.

[2] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, “ Network flows: Theory, Algorithms and Applications”, *Englewood Cliffs*: Prentice Hall, 1993.

[3] I. F. Akyildiz, X. Wang and W. Wang, “ Wireless mesh networks: A survey”, *Computer Networks*,Vol 47, No.4, 2005, pp. 445-487.

[4] M. Alichery, R. Bhatia and L. Li, “Joint channel assignment and routing for throughput optimization in multiradio wireless Mesh networks”, *in Proc. Conf. Mobicom*, 2005.

[5] K. Anand and V. Sharma, “Computing TCP throughput in UMTS Network”, to be presented in WCNC 2008.

[6] A. Anil Kumar, “Providing QoS to Real and Interactive Data Applications in WiMAX Mesh Networks”, *M.E. Thesis, Dept of ECE, IISc, Bangalore* June 2007.

[7] M. Cao, V. Raghunathan and P.R. Kumar, “ A tractable algorithm for fair and efficient uplink scheduling of multi-hop WiMAX mesh networks”, *Preprint, Proceedings of WiMesh 2006: Second IEEE Workshop on Wireless Mesh Networks*, Sept. 2006, Reston, VA.

[8] 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”, *ACM MobiHoc*, 2005.

[9] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance”, *IEEE/ACM Trans. Networking*, Vol. 1, 1993, 397-413.

[10] L. Georgiadis, M.J. Neely and L. Tassiulas, “ Resource allocation and cross-layer control in wireless networks”, *Foundations and Trends in Networking*, 2006, Vol 1, No.1, pp.1-144, Now Publishers.

[11] A. Gupta and V. Sharma, “A Unified approach for analyzing persistent, non-persistent and ON-OFF TCP sessions in the Internet”, *Performance Evaluation*, Vol. 63, 2006, 79-98.

[12] R. Gupta, J. Musacchio and J. Walrand, “ Sufficient rate conditions for QoS flows in adhoc networks”, *Preprint*, 2004.

[13] V. Kamble and V. Sharma, “ A simple approach to provide QoS and fairness in Internet”, in *Proc. International Conf. on Signal Processing and Communications (SPCOM 2004)*, Bangalore, Dec. 2004.

[14] D. Kim and A. Ganz, “ Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems”, *in IEEE Broadnets’05* Boston, 2005.

[15] M. Kodialam and T. Nandagopal, “Characterizing achievable rates in wireless multi-hop networks: The joint routing and scheduling problem”, *ACM Mobicom*, September 2003.

[16] M. Kodialam and T. Nandagopal, “Characterizing the capacity region in multi-radio, multichannel wireless mesh networks”, *in Proc. ACM Mobicom*, 2005.

[17] X. Lin, N. B. Shroff and R. Srikanth, “A tutorial on cross-layer optimization in wireless networks”, *IEEE Journal on Selected Areas in Communication*, Vol 24, August 2006, pp. 1-12.

[18] J. Pandhye, V. Firoiu, D. Towsley and J. Kurose, “Modelling TCP throughput : A simple model and its empirical validation”, *ACM Proc. SIGCOMM’98*, 1998.

[19] V. Reddy, V. Sharma and M. B. Suma, “Providing QoS to TCP and real time connections in the Internet”, *Queueing Systems*, Vol. 46, 2004, 461-480.

[20] S. R. Sandeep, “Joint routing and scheduling for multihop wireless networks”, *M.E. Thesis, Dept of ECE, IISc, Bangalore* June 2006, (Also presented in poster session of *Conference on Stochastic Networks*, Urbana, IL, June 2006).

[21] V. Sharma and P. Punyaslok, “ Stability and analysis of TCP connections with RED control and exogenous traffic”, *Queuing Systems*, Vol 48, 2004, 193-235.

[22] H. Shetiya, “ Efficient routing and scheduling algorithms for IEEE 802.16 mesh networks”, M.E. Thesis, Dept. of ECE, Indian Institute of Science, Bangalore, 2005.

[23] H. Shetiya and V. Sharma, “Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks”, *in Proc. of 1st ACM worshop WMuNeP 2005 in 7th ACM/IEEE MSWiM 2005*, October 2005.

[24] H. Shetiya and V. Sharma, “Algorithms for routing and centralized scheduling in IEEE 802.16 Mesh Networks”, *in Proc IEEE Wireless Communications and Networking Conference (WCNC06)*, April 2006.

[25] J. Walrand and P. Varaiya, “High Performance Communication Networks”, *Morgan Kaufmann*, San Francisco , 1996.

[26] H-Y. Wei, S. Ganguly, R. Izmailov and Z. J. Haas, “ Interference aware IEEE 802.16 WiMAX mesh networks”, *in 61th IEEE Vehicular Technology Conf., (VTC 2005)*, Stockholm, 2005.