Computer Communications 32 (2009) 511–521

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A performance study of uplink scheduling algorithms in point-to-multipoint WiMAX networks

Najah Abu Ali [a,*], Pratik Dhrona [b], Hossam Hassanein [c]

[a] College of IT, UAE University, Al-Ain, United Arab Emirates
[b] Redline Communication, Toronto, Ont., Canada
[c] School of Computing, Queen's University, Kingston, Ont., Canada

ARTICLE INFO

Article history:
Available online 18 September 2008

Keywords:
WiMAX
RRM
Scheduling
OFDM
Performance evaluation

ABSTRACT

The IEEE 802.16 standard defines the specifications for medium access control (MAC) and physical (PHY) layers of WiMAX networks. A critical part of the MAC layer specification is packet scheduling, which resolves contention for bandwidth and determines the transmission order of users. Evaluating the performance packet scheduling algorithms is of utmost importance towards realizing large-scale WiMAX deployment. In this paper, we conduct a comprehensive performance study of scheduling algorithms in point-to-multipoint mode of OFDM-based WiMAX networks. We first make a classification of WiMAX scheduling algorithms, then simulate a representative number of algorithms in each class taking into account that vital characteristics of the IEEE 802.16 MAC layer and OFDM physical layer. We evaluate the algorithms with respect to their abilities to support multiple classes of service, providing quality of service (QoS) guarantees, fairness amongst service classes and bandwidth utilization. To the best of our knowledge, no such comprehensive performance study has been reported in the literature. Simulation results indicate that none of the current algorithms is capable of effectively supporting all WiMAX classes of service. We demonstrate that an efficient, fair and robust scheduler for WiMAX is still an open research area. We conclude our study by making recommendations that can be used by WiMax protocol designers.

1. Introduction

The vast increase in contemporary applications with varying QoS requirements creates a demand for a unified service and/or networking platform that can simultaneously support such applications. To this end, the IEEE 802.16-2004 standard [1] specifies a service framework of four scheduling class services, unsolicited grant service (UGS), real-time polling service (rtPS), non-real time polling service (nrtPS) and best effort (BE). These classes are considerably diverse with respect to bandwidth request/grant mechanism and QoS requirements. For example, to support UGS flows, the base station (BS) is required to allocate fixed size data grants, based on a fixed data rate requested by subscriber stations (SSs) of the UGS class. The IEEE 802.16e-2005 [2] standard introduces an additional scheduling service, extended real-time polling service (ertPS), which builds on the efficiency of both UGS and rtPS. Just like the UGS class, the BS is allowed to provide unicast grants to the ertPS SSs, but the size of the grants can vary, enabling a more efficient usage of available bandwidth.

Although the IEEE 802.16-2004 standard specifies a service framework and its associated bandwidth request/grant mechanisms over single-carrier or multiple-carrier physical layer technologies such as OFDM and OFDMA, it does not specify the scheduling algorithm to allocate the OFDM or OFDMA frame symbols that enforce QoS requirements of all traffic classes. Accordingly, there have been several proposals of scheduling algorithms, some based on legacy algorithms [3–9]. Others are designed specifically for WiMAX and some are tailored to WiMAX standard specifications [10–14], WiMAX specific algorithms are centered on the major characteristics of the MAC layer, as specified by the IEEE 802.16-2004 standard.

Despite the numerous scheduling algorithms proposed for WiMAX networks, there is no comprehensive study that provides a unified platform for comparing such algorithms. The aim of this work is to allow a thorough understanding of the relative performance of representative uplink scheduling schemes and subsequently utilize the results to address their scarcity in designing more efficient schemes. We focus our work on implementing representative algorithms for the uplink traffic in OFDM WiMAX physical layer using network simulator 2 (NS-2). Another major contribution of this work is evaluating the algorithms using traffic models specifically designed for WiMAX to represent its diverse

[*] Corresponding author.
E-mail addresses: najah@uaeu.ac.ae (N.A. Ali), pratik@cs.queensu.ca (P. Dhrona), Hossam@cs.queensu.ca (H. Hassanein).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.09.015

applications [19], incorporating the mandatory and some optional parameters of all the traffic classes as specified in the IEEE 802.16-2004 standard. The traffic model has also been implemented in NS-2. It is our intention to add our contributions to the NS-2 public code base, which may assist future evaluations and make it easier to implement variations of the evaluated scheduling schemes. The findings of this work are beneficial both for academia and industry. For instance, our analysis indicates that none of the current algorithms is capable of providing efficient, fair, and robust scheduler to support all the WiMAX classes. As well, the analysis and conclusions from this study can be used in understanding the strengths and weaknesses of current scheduling algorithms and thus designing efficient scheduling algorithms that addresses some or all of these weaknesses. The study is also beneficial to WiMAX BSs and SSs manufacturers, vendors and operators in that it provides a baseline to choose appropriate scheduling algorithms depending on the traffic profiles in their network.

The rest of the paper is organized as follows. In Section 2, a survey of scheduling algorithms for the uplink traffic in WiMAX is provided. This section also includes justification of selecting representative algorithms for our simulation study. Section 3 describes the simulation framework that includes the simulation parameters, traffic model, and the performance metrics used to evaluate the algorithms and the results and discussion of the experiments. In Section 4, we summarize the results, provide suggestions for improvement and discuss some future research directions.

2. Scheduling algorithms in PMP WiMAX networks

Packet scheduling is the process of resolving contention for bandwidth. A scheduling algorithm has to determine the allocation of bandwidth among the users and their transmission order. One of the most important tasks of a scheduling scheme is to satisfy the quality of service (QoS) requirements of its users while efficiently utilizing the available bandwidth. For the uplink traffic, the scheduling algorithm has to work in tandem with call admission control (CAC) to satisfy the QoS requirements. The CAC algorithm ensures that a connection is accepted into the network only if its QoS requirements can be satisfied as well as the performance of existing connections in the network is not deteriorated.

In our survey, several scheduling algorithms are assessed with respect to the characteristics of the IEEE 802.16 MAC layer and OFDM physical layer. We classify the proposals into three categories; homogenous algorithms, hybrid algorithms and opportunistic algorithms. Homogenous and hybrid categories consist of legacy algorithms with the hybrid category employing multiple legacy schemes in an attempt to satisfy the QoS requirements of the multi-class traffic in WiMAX networks. The opportunistic category refers to algorithms that exploit variations in channel conditions in WiMAX networks whilst incorporating the QoS requirements in their scheduling design. Representative schemes in each of these categories will be discussed next.

2.1. Homogeneous algorithms

Weighted Round Robin (WRR) and Deficit Round Robin (DRR) algorithms are evaluated in a WiMAX network in reference [3]. WRR is evaluated for the uplink traffic while DRR is evaluated for the downlink traffic. In WRR, weight to each SS can be assigned to reflect their relative priority. Priority of the SSs can also be incorporated in the DRR algorithm. DRR allows provision of different quanta for each SS. A higher quantum can be assigned to higher priority SSs. Ruangchaijatupon et al. [4] evaluate the performance of Earliest Deadline First (EDF) algorithm. EDF is a work conserving algorithm originally proposed for real-time applications in wide area networks [5]. The algorithm assigns deadline to each packet and allocates bandwidth to the SS that has a queued packet with the earliest deadline. Weighted fair queuing (WFQ) is also evaluated and compared with EDF in reference [4].

Tsai et al. [6] propose an uplink scheduling algorithm and a token bucket based Call Admission Control (CAC) algorithm. The CAC algorithm assigns thresholds to each class to avoid starvation of lower priority classes. The scheduling algorithm first grants bandwidth to SSs of the UGS class. The algorithm then allocates bandwidth to SSs of the rtPS class using EDF algorithm and restricting the allocation to the maximum grant size. Finally, the algorithm allocates minimum required bandwidth to SSs of the nrtPS and BE classes, in that order.

2.2. Hybrid algorithms

Wongthavarawat and Ganz [7] propose a hybrid scheduling algorithm that combines EDF, WFQ and FIFO scheduling algorithms. The overall allocation of bandwidth is done in a strict priority manner. EDF scheduling algorithm is used for SSs of the rtPS class, WFQ is used for SSs of the nrtPS class and FIFO for SSs of the BE class. Besides the scheduling algorithm, an admission control procedure and a traffic policing mechanism are also proposed.

Vinay et al. [8] propose a hybrid scheme that uses EDF for SSs of the rtPS class and WFQ for SSs of nrtPS and BE classes. This algorithm differs from [7] in that WFQ is used for SSs of both nrtPS and BE classes and the overall bandwidth is allocated fairly, however, the authors did not describe the mechanism for fair allocations. Settembre et al. [9] propose a hybrid scheduling algorithm that uses WRR and RR algorithms with a strict priority mechanism for overall bandwidth allocation. In the initial stages, bandwidth is allocated on a strict priority basis to SSs of the rtPS and nrtPS classes only. The WRR algorithm is used to allocate bandwidth amongst SSs of rtPS and nrtPS classes until they are satisfied. Any residual bandwidth is distributed between the SSs of the BE class using the RR algorithm.

A vital component of hybrid algorithms is the distribution of bandwidth among the diverse traffic classes. We have selected to evaluate hybrid (EDF + WFQ + FIFO) and hybrid (EDF + WFQ) schemes, which use very different mechanisms of distributing bandwidth among the traffic classes. The hybrid (EDF + WFQ + FIFO) algorithm applies the strict priority mechanism, whereas the hybrid (EDF + WFQ) keeps track of the bandwidth allocated to all the service classes and perform dynamic distribution of bandwidth by providing fair service to all the traffic classes. In our evaluation, we use the Minimum Reserved Traffic Rate (MRTR) of a SS as the core of this fair approach (details were not available in [8]). More specifically, bandwidth is distributed with respect to the relative MRTR of all SSs in a class, i.e. the available bandwidth is multiplied by the ratio of sum of MRTR of SSs in a class to the sum of MRTR of all the SSs in the network.

2.3. Opportunistic algorithms

A Cross-Layer scheduling algorithm is proposed in reference [10] whereby each SS is assigned a priority based on its channel quality and service status. The SS with the highest priority is scheduled for transmission in each frame. The algorithm considers all the required QoS parameters of the scheduling services specified in the IEEE 802.16-2004 standard. Class coefficients are utilized to assign relative priority to the different traffic classes. Rath et al. [11] propose to use an opportunistic extension of the Deficit Round Robin (DRR) algorithm with the purpose of satisfying delay requirements of multi-class traffic in WiMAX. The heart of the algorithm lies in selecting an appropriate polling algorithm. At the beginning of a polling interval, a set of schedulable SSs are selected that constitute a schedulable set. Until the next polling interval, SSs are selected only from the schedulable set.

Niyato and Hossain [12] propose a joint bandwidth allocation and connection admission control algorithm based on queuing theory. In order to limit the amount of bandwidth allocated per class, a bandwidth threshold is assigned to each class. A utility function is calculated for each SS based on the QoS requirements of the traffic class. Subsequently, bandwidth is allocated based on the utility, giving priority to the SS with the lowest utility.

Singh and Sharma [13] propose a scheduling algorithm for OFDMA systems with a TDD frame structure for both uplink and downlink traffic in WiMAX. The algorithm allocates bandwidth among the SSs on a priority basis taking into consideration the channel quality, the number of slots allotted to the SS and the total bandwidth demanded by the SS. Kim and Yeom propose an uplink scheduling algorithm for TCP traffic for the BE class [14]. The proposed algorithm does not require explicit bandwidth request from a SS. It estimates the amount of bandwidth required by the SS based on its current sending rate. The purpose of the algorithm is to provide reasonable fairness among the SSs based on the min-max fairness criteria while providing high frame utilization.

The Cross-Layer and Queuing Theoretic algorithms provide a good representation of all the schemes in this category. Both algorithms differ with respect to the number of SSs selected for transmission and the QoS parameters incorporated. The Queuing Theoretic algorithm schedules multiple SSs in each frame whereas the Cross-Layer algorithm schedules only one SS. The Cross-Layer algorithm includes both throughput and delay in the priority function of rtPS class but the Queuing Theoretic algorithm includes only the delay in the utility function of the rtPS class.

2.4. Uplink scheduling algorithms under evaluation

In this section, we provide implementation details of the representative uplink scheduling algorithms in WiMAX. As part of the implementation details, we also highlight any assumptions and implementation decisions made in the process.

Following are variables/functions used in the pseudo-code for the scheduling algorithms:

- C: the uplink channel capacity.
- Ω_{Total}: the set of all admitted SSs.
- Ω_{ertPS}: set of all SSs belonging to the ertPS class.
- Ω_{rtPS}: set of all SSs belonging to the rtPS class.
- Ω_{nrtPS}: set of all SSs belonging to the nrtPS class.
- Ω_{BE}: set of all SSs belonging to the BE class.
- b_i^{alloc}: bandwidth allocated to SS *i*.
- deque_i(P): remove packet *P* from the queue of SS *i*.
- enque_i(P): insert packet *P* in the queue of SS *i*.
- size_i(P, γ_i): retrieve the size of packet *P* from the queue of SS *i*. Convert the size of the packet to number of symbols based on the signal to interference noise ratio (SINR (γ_i)), of SS *i*.
- P_i^{HOL}: head of line (HOL) packet of SS *i*.
- CreateIE (size_i(P, γ_i), *i*): create an information element (IE) for SS *i* with size_i(*P*, γ_i) number of symbols. After creation, the information element (IE) is added to the ULMAP message.
- Next_i(P): retrieve the next packet from the queue of SS *i*.
- drop (ertPS, rtPS): drop packets from the queue of all SSs of ertPS and rtPS class.

2.4.1. Homogenous scheduling algorithms

The WRR scheduling algorithm originally proposed for ATM traffic has been implemented in [3] to evaluate the IEEE 802.16 MAC layer on how effectively it supports QoS requirements of the multi-class traffic (Algorithm-1). A critical portion of the WRR scheme is assigning weights to each SS. The weights are assigned to reflect the relative priority and QoS requirements of the SSs. Since MRTR is one of the parameters specified by a SS that reflect its QoS requirements, we assign weight to each SS with respect to its MRTR as follows:

$$W_i = \text{MRTR}_i \Bigg/ \sum_{j=1}^{n} \text{MRTR}_j \tag{2.1}$$

where, W_i is the weight of SS *i*. *n* the number of SSs.

Algorithm 1. Pseudo-code of WRR algorithm

```
drop(ertPS, rtPS)
Assign weights to SSs using (2.1).
while($C > 0$)
    for $i \in \Omega_{\text{Total}}$ do
        $b_i^{\text{alloc}} = b_i^{\text{alloc}} + W_i * C$
        $C = C - W_i * C$
    end for
end while
for$i \in$ Conn do
    CreateIE() //create IEs based on allocation bandwidth item   end for
```

EDF is one the most widely used scheduling algorithms for real-time applications as it selects SSs based on their delay requirements. The algorithm assigns deadline to arriving packets of a SS (Algorithm-2). Since each SS specifies value for the maximum latency parameter, the arrival time of a packet is added to the latency to form the tag of the packet. The value of maximum latency for SSs of the nrtPS and BE classes is set to infinity. In the following pseudo-code, $\text{min}_{\text{deadline}}$(P) refers to the packet with the earliest deadline.

Algorithm 2. Pseudo-code of EDF algorithm

```
drop(ertPS, rtPS)
Assign deadline upon arrival of a packet.
while($C > 0$)
    $b_\text{i}^{\text{alloc}} = b_\text{i}^{\text{alloc}} + \text{size}_\text{i}(\text{min}_{\text{deadline}}(P), \gamma_i)$
    CreateIE()
    $C = C\text{-size}_\text{i}(\text{min}_{\text{deadline}}(P), \gamma_\text{i})$
end while
```

Both WFQ and WRR scheduling algorithms assign weights to SSs. Unlike the WRR algorithm, the WFQ algorithm also considers the packet size and the channel capacity when allocating bandwidth to the SSs (Algorithm-3). An arriving packet is tagged with finish time that is calculated based on the weight of the SS, the packet size and the uplink channel capacity. In WFQ, weight of a SS is calculated in the same way as it is in WRR. Once the weight is assigned, arriving packets of the SS are stamped with virtual finish time as follows:

$$S_k^i = \max\{F_i^{k-1}, V(a_i^k)\} \tag{2.2}$$

$$F_i^k = S_k^i + L_k^i/\phi_i \tag{2.3}$$

where, S_k^i is the start time of *k*th packet of SS *i*.
F_i^{k-1} is the finish time of $(k-1)$th packet of SS *i*.
$V(a_i^k)$ is the virtual time of the *k*th packet of SS *i*.
a_i^k arrival time of *k*th packet of SS *i*.
F_i^k is the finish time of kth packet of SS *i*.
L_k^i is the length of *k*th packet of SS *i*.
ϕ_i is the reserved rate of SS *i*, where $\phi_i = C^* W_i$
W_i is the weight assigned to SS *i*.

The complexity of WFQ is high due to two main reasons; selecting the next queue to serve and computation of the virtual time. The complexity of the former is fixed to O (log$N$) where as the complexity of the latter is *O(N),* where N is the number of SSs. The transmission of a packet will trigger an update to the virtual time of its SS. Since the scheduling algorithm is implemented at the BS, the virtual time will be updated once a packet has been selected for transmission. For a time interval τ, virtual time $V(t)$ is updated as follows:

$$V(t_{j-1}+\tau) = V(t_{j-1}) + \frac{\tau}{\sum_{i\in B_j}\phi_i} \tag{2.4}$$

where, $\tau \leqslant t_j - t_{j-1} j = 2,3,\ldots$

B_j is the set of busy SSs.

Algorithms 3-1 and 3-2 describe the pseudo-code of the actions that need to be performed upon arrival and selection of a packet, respectively. In the following pseudo code, $\min_{\text{finishtime}}$(P) refers to the packet with the smallest finish time.

Algorithm 3-1. Action upon arrival of packet k of SS i – arrive(i,k)

```
if system idle
   V(t)=0
   $F_i^k = 0$
end if
Calculate $S_k^i$ and $F_i^k$ using (2.2) and (2.3)
if $i \in B$ then
   $B = B + i$        //Add SS $i$ to the busy set
end if
assign $F_i^k$ to packet $k$
```

Algorithm 3-2. Action upon selection of packet k of SS i – select (i,k)

```
Update $V(t)$ using (2.4)
CreateIE()
if connection i not backlogged
   $B = B - i$        //Remove $i$ from the busy set
end if
```

Algorithm 3. Pseudo-code of WFQ algorithm

```
drop(ertPS, rtPS)
Upon arrival of packet $k$ of SS $i$,
   arrive(i,k)
enque$_i$(k)
while($C > 0$)
   $b_i^{\text{alloc}} = b_i^{\text{alloc}} + \text{size}_i(\min_{\text{finishtime}}(P), \gamma_i)$
   $C = C - \text{size}_i(\min_{\text{finishtime}}(P), \gamma_i)$
   select($i$,$k$)
end while
```

2.4.2. Hybrid algorithms

The hybrid algorithm proposed in [7] uses strict priority mechanism for overall bandwidth allocation (algorithm-4). EDF scheduling algorithm is used for SSs of ertPS and rtPS classes, WFQ algorithm is used for SSs of nrtPS class and FIFO is used for SSs of BE class. The EDF and WFQ algorithms are implemented as described above. FIFO is used for BE class as SSs of this class do not have any QoS requirements. In the following pseudo-code, queue(Conn$_{\text{ertPS}}$), queue (Conn$_{\text{rtPS}}$), queue(Conn$_{\text{nrtPS}}$) and queue(Conn$_{\text{BE}}$) refer to packet queues of SSs from ertPS, rtPS, nrtPS and BE classes, respectively.

Algorithm 4. Pseudo-code of hybrid (EDF+WFQ+FIFO) algorithm

```
drop(ertPS,rtPS)
Upon arrival of packet $k$ of connection $i$
   if($i \in$ Conn$_{\text{ertPS}}$, Conn$_{\text{rtPS}}$)
      Assign deadline to packet $k$
   end if
   if($i \in$ Conn$_{\text{nrtPS}}$)
      arrive($i$,$k$)
   end if
enque$_i$($k$)
for $i \in$ Conn$_{\text{ertPS}}$, Conn$_{\text{rtPS}}$
   while($C > 0$ and (queue(Conn$_{\text{ertPS}}$) or queue(Conn$_{\text{rtPS}}$))!=NULL)
      $b_i^{\text{alloc}} = b_i^{\text{alloc}} + \text{size}_i(\min_{\text{deadline}}(P), \gamma_i)$
      CreateIE()
      $C = C - \text{size}_i(\min_{\text{deadline}}(P), \gamma_i)$
   end while
end for
for $i \in$ Conn$_{\text{nrtPS}}$
   while($C > 0$ and queue(Conn$_{\text{nrtPS}}$)!=NULL)
      $b_i^{\text{alloc}} = +\text{size}_i(\min_{\text{finishtime}}(P), \gamma_i)$
      $C = C - \text{size}_i(\min_{\text{finishtime}}(P), \gamma_i)$
      select(i,k)
   end while
end for
while($C > 0$ and queue(Conn$_{\text{BE}}$)!=NULL)
   $b_i^{\text{alloc}} = b_i^{\text{alloc}} + \text{deque}_i(P)$
   CreateIE()
   $C = C - \text{size}_i\ (P,\gamma_i)$
end while
```

Vinay et al. [8] propose a hybrid algorithm that uses EDF scheduling algorithm for SSs of ertPS and rtPS classes and WFQ algorithm for SSs of nrtPS and BE classes (Algorithm-5). Although the mechanism of overall bandwidth distribution is not specified, it is mentioned in reference [8] that bandwidth is allocated in a fair manner. The following is the overall bandwidth allocation scheme adopted in our implementation:

$$\text{BW}_{\text{ertPS,rtPS}} = C * \left(\sum_{i\in\text{ertPS,rtPS}} \text{MRTR}_i\right) \Bigg/ \left(\sum_{j=1}^{n} \text{MRTR}_j\right) \tag{2.5}$$

$$\text{BW}_{\text{nrtPS,BE}} = C * \left(\sum_{i\in\text{nrtPS,BE}} \text{MRTR}_i\right) \Bigg/ \left(\sum_{j=1}^{n} \text{MRTR}_j\right) \tag{2.6}$$

Algorithm 5. Pseudo-code of hybrid (EDF + WFQ) algorithm

```
drop(ertPS,rtPS)
Assign overall bandwidth using (2.5) and (2.6)
Upon arrival of packet $k$ of SS $i$
  if($i \in \Omega_{ertPS}, \Omega_{rtPS}$)
    Assign deadline to packet k
  end if
  if($i \in \Omega_{nrtPS}, \Omega_{BE}$)
    arrive($i$,$k$)
  end if
enque$_i$($k$)
for $i \in \Omega_{ertPS}, \Omega_{rtPS}$
  while ($BW_{ertPS,rtPS} > 0$ and (queue($\Omega_{ertPS}$) or queue($\Omega_{rtPS}$))!=NULL)
    $b_i^{alloc} = b_i^{alloc} + size_i(min_{deadline}(P), \gamma_i)$
    CreateIE()
    $BW_{ertPS,rtPS} = BW_{ertPS,\ rtPS} - size_i(min_{deadline}(P), \gamma_i)$
  end while
end for
if($BW_{ertPS,rtPS} > 0$)
  $BW_{nrtPS,BE} = BW_{nrtPS,\ BE} + BW_{ertPS,rtPS}$
end if
for $i \in \Omega_{nrtPS}, \Omega_{BE}$
  while ($BW_{nrtPS,BE} > 0$ and (queue($\Omega_{nrtPS}$) or queue($\Omega_{BE}$))!=NULL)
    $b_i^{alloc} = b_i^{alloc} + size_i(min_{finishtime}(P), \gamma_i)$
    CreateIE()
    $BW_{nrtPS,BE} = BW_{nrtPS,\ BE}- size_i(min_{finishtime}(P), \gamma_i)$
    select($i$,$k$)
  end while
end for
```

2.4.3. Opportunistic algorithms

The two opportunistic algorithms, Cross-Layer [10] and Queuing Theoretic [12] selected for evaluation use priority and utility functions in calculating the relative priority of the SSs. The Queuing Theoretic algorithm uses queuing theory and sigmoid function in assigning utility to the SSs whereas the Cross-Layer algorithm uses delay of packet and Average Throughput. Detailed explanation of the algorithms can be found in [10,12].

2.5. Computational complexity

The computational complexity of a scheduling algorithm strongly influences its scalability. The complexities of homogenous algorithms are well documented in the literature and mentioned below for completeness. However, and to the best of our knowledge, the complexities of hybrid and opportunistic algorithms have not been reported. In the discussion that follows, N refers to the number of SSs.

The complexity of the WRR algorithm is known to be constant with respect to the number of SSs, i.e. $O(1)$ [15]. It has been discussed in [16] that the complexity of the WFQ algorithm is *O(N)*, where *N* is the number of SSs. The complexity of the EDF algorithm is also *O(N)* [17]. The complexity of the hybrid (EDF + WFQ + FIFO) and hybrid (EDF + WFQ) algorithms is *O (N),* per their legacy components.

The complexity of the Cross-Layer algorithm is $O(N)$. Its complexity is dominated by the portion of the algorithm that assigns priority to the SSs that loops through all the SSs. The complexity of the Queuing Theoretic algorithm is dictated by the portion of the algorithm that assigns one unit of bandwidth to the SS in a round robin fashion. This results in the overall complexity of the algorithm to be $O(N^2)$.

3. Simulation framework and experiments

We have extended the NS-2 module for WiMAX PMP mode [18], version 1.06, for the simulation experiments. Two major contributions have been augmented to the extension. The first contribution is the integration of seven representative uplink scheduling algorithms and modification to the bandwidth management function so that the list of Information Elements (IEs) in the ULMAP message is created according to the transmission order determined by the scheduling algorithm. The second major contribution is the addition of a WiMAX traffic model based on [19] that implements voice over internet protocol (VoIP) traffic for the ertPS class, video streaming traffic for the rtPS class, FTP traffic for the nrtPS class and HTTP traffic for the BE class.

It is assumed that the BS has channel state information of all SSs. A SS conveys its received signal to noise power ratio (SNR) from itself to the BS through a robust feedback channel. This information is not changing over one time frame, since the wireless channel between each SS and BS is assumed to undergo a flat fading that is fixed over one frame period. The wireless channel is modeled based on Nakagami-m channel model which is adopted to accurately describe the statistical variation of the channel gains between the BS and the SSs based on OFDM channel multiplexing.

The adaptive modulation coding (AMC) module defined by the IEEE 802.16 standard divides the range of the received SNR into seven non-overlapping regions where the dividing thresholds are evaluated based on a target prescribed bit error rate (BER). According to the received SNR at a specific SS, the module in the PHY layer of BS decides the suitable transmission mode for each SS as shown in Table 1. Each transmission mode consists of modulation and coding pair aims at efficiently using the bandwidth while satisfying a prescribed BER.

3.1. Traffic model and simulation parameters

3.1.1. Traffic model

We have implemented four different traffic sources, one for each WiMAX traffic class. VoIP traffic is modeled for SSs of ertPS class, video streaming for SSs of rtPS class, FTP for SSs of nrtPS class and HTTP for SSs of BE class. The values of all the traffic parameters are based on one connection per SS. The FTP traffic model consists of a constant packet size of 150 bytes, a Minimum Reserved Traffic Rate (MRTR) of 45 kbps and a Maximum Sustained Traffic Rate (MSTR) of 500 kbps. The HTTP traffic model consists of a constant packet size of 100 bytes with an MSTR of 64 kbps. Parameters of the VoIP model based on Adaptive Multi-Rate (AMR) coding and video streaming traffic model are listed in Tables 2 and 3, respectively.

3.1.2. Simulation parameters

We consider a single cell consists of one BS with several SSs in a PMP mode. TDD multiplexing is used to divide transmission time frame into equal UL and DL sub-frames. The UL is shared between

Table 1
IEEE 802.16-2004 transmission modes

Mode	1	2	3	4	5	6
Modulation	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM
Coding rate	1/2	3/4	1/2	3/4	2/3	3/4
Raw bit rate (Mbps)	15.36	23.04	30.72	46.08	61.44	69.12

Table 2
VoIP traffic parameters

Parameter	Value (1 connection per SS)
Minimum Reserved Traffic Rate (MRTR)	25 kbps
Average traffic rate	44 kbps
Maximum Sustained Traffic Rate (MSTR)	64 kbps
Maximum latency	100 ms
Tolerated Packet loss	10%
Talk spurt length	Exponential random, $\mu = 147$ ms
Silent period length	Exponential random, $\mu = 167$ ms
Packet size	23 bytes

Table 3
Video streaming parameters

Parameter	Value (1 connection per SS)
Minimum Reserved Traffic Rate (MRTR)	64 kbps
Average traffic rate	282 kbps
Maximum Sustained Traffic Rate (MSTR)	500 kbps
Maximum latency	150 ms
Tolerated packet loss	5%
Packet size	150–300 bytes

SSs implementing Time Division Multiple Access (TDMA) mechanism over OFDM PHY layer. The DL bandwidth is simulated as 20 MHz. Nodes are placed in random over a simulation grid of 1000 m × 1000 m. Number of subscriber stations varies from 6 to 30. Each subscriber station can have connections of one type of traffic at the same time; i.e. voice, video, FTP or HTTP traffic. The ratio of connections in the network changes from 1 to 3. For example if the ratio is 1:1:1:1, then number of connections in the network for each type are equal. The frame size is fixed during the simulation time and it varies from 4 to 20 ms for different simulation scenarios equally divided between UL and DL traffic, the symbol duration is 12.5 μs. Time slots – which are directly related to OFDM symbols – are allocated for active flows at the beginning of each frame, where one connection is the maximum number of connections allocated per one time slot.

3.1.3. Performance metrics

Average Throughput: The amount of data transmitted by a user over the simulation time. This metric is used to evaluate the performance of the opportunistic schedulers and to understand the effect of preamble overhead and the relative priority given to the SSs.

Average queuing delay: The time between the arrival of a packet to the departure of the packet from the queue. The value is reported in milliseconds (ms) and is averaged over number of packets.

Packet loss: The percentage of packets dropped from the queue out of all the packets that arrived into the queue. The metric indicates the percentage of packets that missed their delay bounds. Overflow of the queue due to violating the Service Level Agreement (SLA) of flows is out of scope of the paper, since we are evaluating the scheduling algorithms and not a traffic policing or conditioning algorithms. Both average delay and Packet loss will allow us to determine how effectively a scheduling algorithm satisfies the QoS requirements of real-time SSs.

Frame utilization: The number of symbols utilized for data out all the symbols in the uplink sub-frame. The metric is used to determine how effectively the scheduling algorithm utilizes the frame.

Fairness index: Fairness is measured between users of the same traffic class (intra-class fairness) and among all users (inter-class fairness). We will use Jain's fairness index to calculate inter-class fairness. Due to the fact that all connections from the same class should receive the same QoS, we use min–max index as it is sensitive to service degradation and service unfairness.

The fairness indices are defined as follows:

$$\text{Jain's fairness index} = \frac{\left(\sum_{i=1^n} x_i\right)^2}{n * \sum_{i=1^n} (x_i)^2}$$

$$\text{Min—Max fairness index} = \frac{x_{\min}}{x_{\max}}$$

To calculate inter-class fairness using Jain's index, we use normalized Average Throughput for x_i. The Average Throughput of a SS is normalized with respect to the MRTR of the SS, i.e. $x = \frac{\hat{\tau}}{MRTR}$. Min–Max fairness calculates fairness between the SS with the maximum Average Throughput ($x_{\max}$) in the class and the SS with the minimum Average Throughput ($x_{\min}$) in the class.

Honored requests: To evaluate the scheduling algorithms with respect to various bandwidth request mechanisms, we calculate the percentage of successful request out of all the requests sent by a SS to the BS. This metric will be referred to as percentage of honored requests in the discussion.

3.2. Simulation results

3.2.1. The effect of traffic type combinations

The goal of this experiment is to study the effect of different combinations of traffic types on the fairness experienced by the SSs. We have measured both intra-class fairness i.e., fairness between SSs of the same class and inter-class fairness i.e., fairness between all the SSs. To highlight differences amongst connections of a same class, we use a sensitive index such as min–max index to measure intra-class fairness [20]. Jain's index is used for inter-class fairness. The mixture of traffic classes is studied by changing the ratio of SSs of the four traffic classes. More specifically, a ratio of 1:2:2:1 with 36 SSs equates to 6 ertPS SSs, 12 rtPS SSs, 12 nrtPS SSs and 6 BE SSs. We adopt a frame length of 20 ms and 36 SSs.

When the concentration of SSs of the nrtPS class is high, the fairness among SSs of the ertPS class under the WFQ algorithm is low (Fig. 3-1(a)). WFQ allocates bandwidth to the SSs based on their MRTR. However, any residual bandwidth is shared among active SSs based on their weights. The weight of the ertPS SSs is low due to their low concentration and low MRTR. Thus, based on the active SSs, packet finish times and residual bandwidth, some SSs may be allocated bandwidth according to their MSTR, while others are allocated bandwidth according to their MRTR. A similar behavior can be observed for the Queuing Theoretic algorithm for both ertPS SSs and rtPS SSs (Fig. 3-1(a) and (b)). However, the difference between the minimum and maximum average throughput is smaller due to the fact that the algorithm allocates the residual capacity on a unit of PDU in a round robin fashion. The hybrid (EDF + WFQ) algorithm shows low intra-class fairness for the BE class when the concentration of nrtPS or BE class is low (Fig. 3-1(c) and (d)). This behavior is due to the competition of BE and nrtPS SSs for the small amount of bandwidth allocated to them as a result of the overall bandwidth allocation mechanism of the algorithm.

The EDF algorithm shows a higher inter-class fairness when the concentration of SSs of the BE class is high but the fairness decreases with increased concentration of SSs of the ertPS, rtPS and nrtPS classes (Fig. 3-1(e)). With large number of SSs of the nrtPS class, the ertPS and rtPS SSs are allocated large amount of bandwidth whereas the nrtPS SSs compete with BE SSs for bandwidth. This results in a significant difference in bandwidth allocated between the different classes. The hybrid (EDF + WFQ + FIFO) algorithm shows high inter-class fairness even when the concentration of ertPS and rtPS SSs is high because it provides strict priority to nrtPS SSs over BE SSs. Since the inter-class fairness is calculated using Average Throughput normalized with respect to

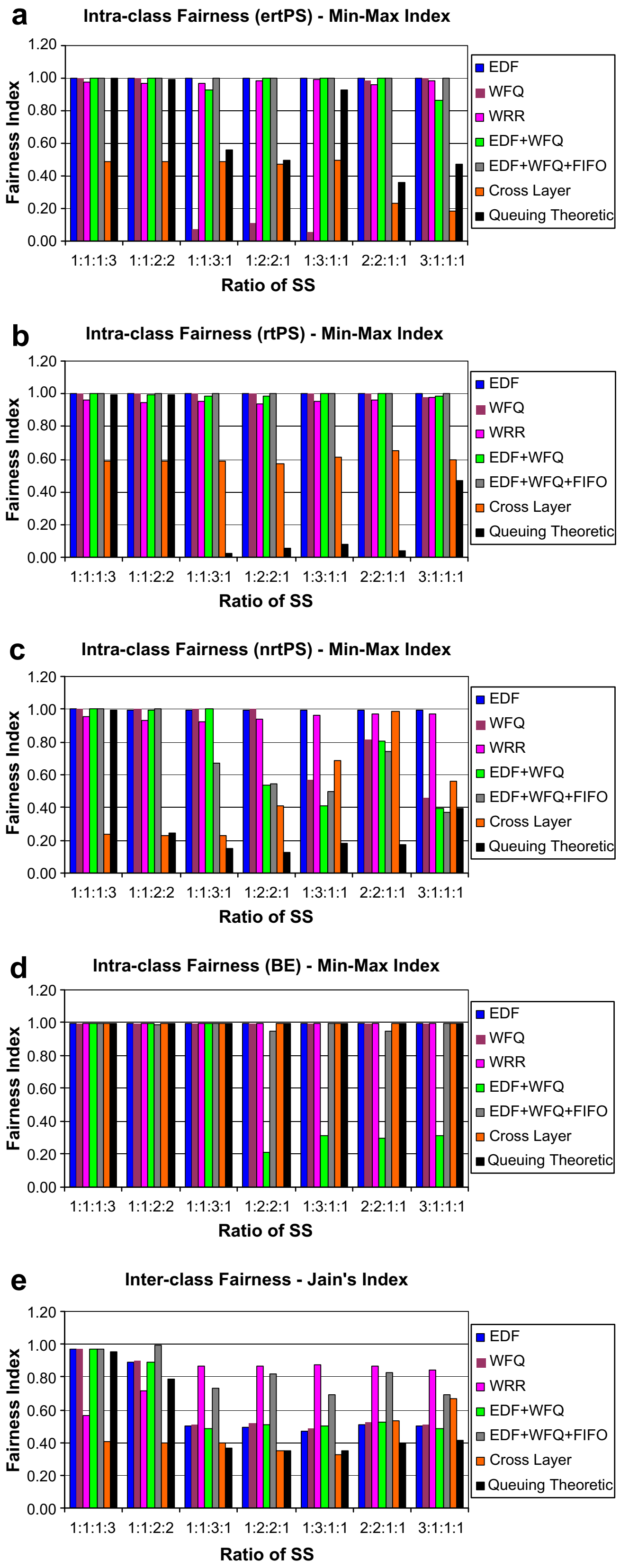

Fig. 3-1. The effect of SS ratio:fairness. (a) Intra-class fairness – ertPS. (b) Intra-class fairness – rtPS. (c) Intra-class fairness – nrtPS. (d) Intra-class fairness – BE. (e) Inter-class fairness.

MRTR of the SSs, the WRR algorithm results in high inter-class fairness. This is because the WRR algorithm assigns weight to the SSs according to their MRTR and serves all the classes in rounds. The Queuing Theoretic algorithm shows low inter-class fairness when the concentration of rtPS SSs is high. Fairness is calculated based on the Average Throughput, but the utility function for rtPS SSs does not take the Average Throughput into consideration, which is an important QoS parameter for the class. Therefore, the unfairness of rtPS SSs due to the Average Throughput is not detected by the utility function, although the purpose of the utility function is to provide fairness.

3.2.2. The effect of uplink burst preamble

The goal of this experiment is to study the effect of preamble symbols that is used by each SS in the uplink direction. According to the IEEE 802.16-2004 [1] standard, the length of the preamble per SS can be either 16 symbols or 32 symbols. In our experiments, we use 16 symbols for uplink preamble. The purpose of the uplink burst preamble is to allow the BS to synchronize with each SS. This can result in a significant overhead when the number of SSs is large, as we will observe from the results of the experiments.

As the number of SSs increases, the frame utilization observed for the Queuing Theoretic and WRR algorithms significantly decreases (Fig. 3-2). With a large number of SSs, the Cross-Layer algorithm shows higher frame utilization than WRR and Queuing Theoretic algorithms. This is because the preamble overhead under the WRR and Queuing Theoretic algorithms is very high resulting from selecting maximum number of SSs in each frame. The frame utilization of the hybrid (EDF + WFQ) algorithm is lower than other legacy algorithms as a result of the bandwidth allocated to each class not being utilized efficiently.

The EDF and hybrid (EDF + WFQ + FIFO) algorithms show higher average throughput for SSs of ertPS and rtPS classes due to their strict priority nature towards real-time SSs and thus resulting in poor performance for nrtPS SSs (Fig. 3-3(a)–(c)). The Cross-Layer algorithm results in low Average Throughput for all the SSs, although the Average Throughput of ertPS SSs relative to other SSs is high. The Cross-Layer algorithm provides very high priority to ertPS SSs due to their tight delay requirement. The Queuing Theoretic algorithm results in high Average Throughput for all the SSs, except under large number of SSs due to the high preamble overhead. The WFQ algorithm shows almost identical Average Throughput for SSs of the rtPS class when compared with EDF and hybrid (EDF + WFQ + FIFO) algorithms. This behavior is caused by the high MRTR of SSs of the rtPS class, allowing the WFQ algorithm to allocate a large amount of bandwidth for them. The WRR algorithm results in very low Average Throughput for the BE class due to the low MRTR of SSs of the class and the large packet size of the BE traffic (Fig. 3-3(d)).

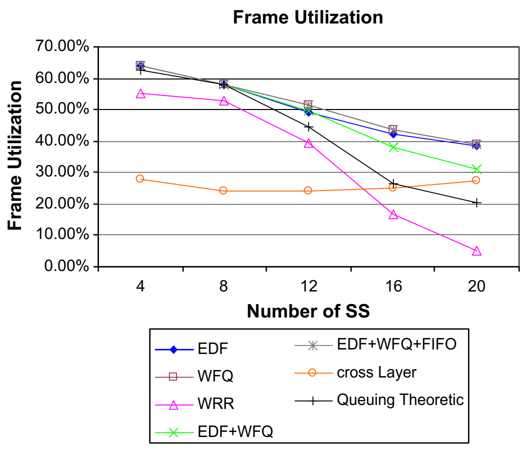

Fig. 3-2. The effect of uplink burst preamble – frame utilization.

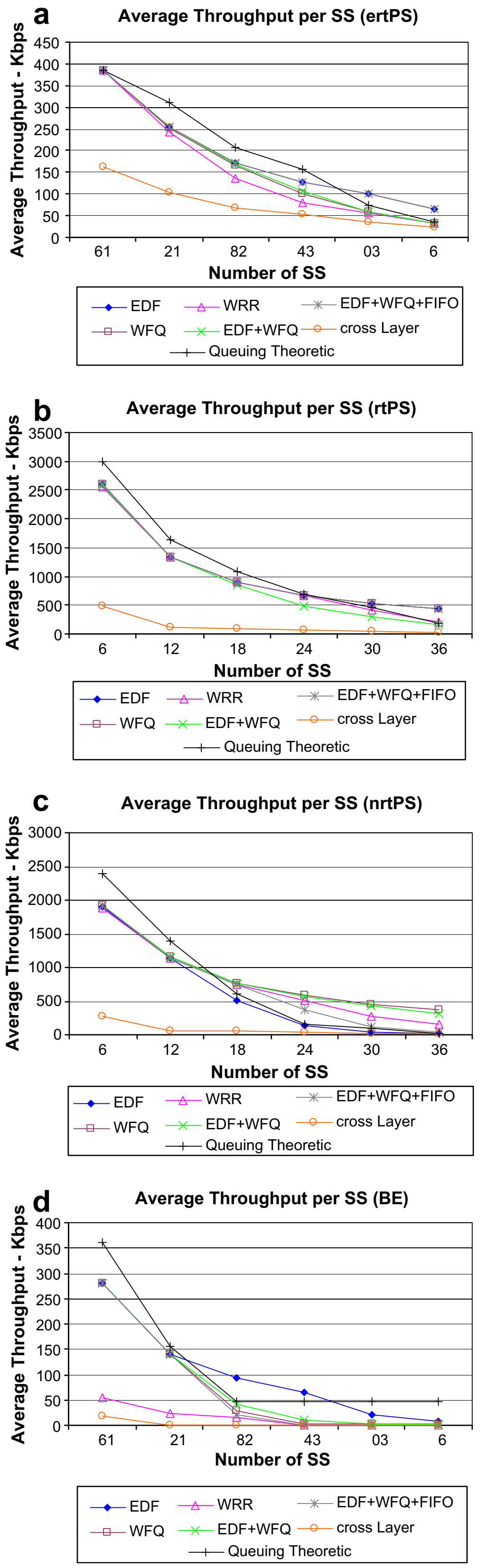

Fig. 3-3. The effect of uplink burst preamble – Average Throughput. (a) Average Throughput – ertPS. (b) Average Throughput – rtPS. (c) Average Throughput – nrtPS. (d) Average Throughput – BE.

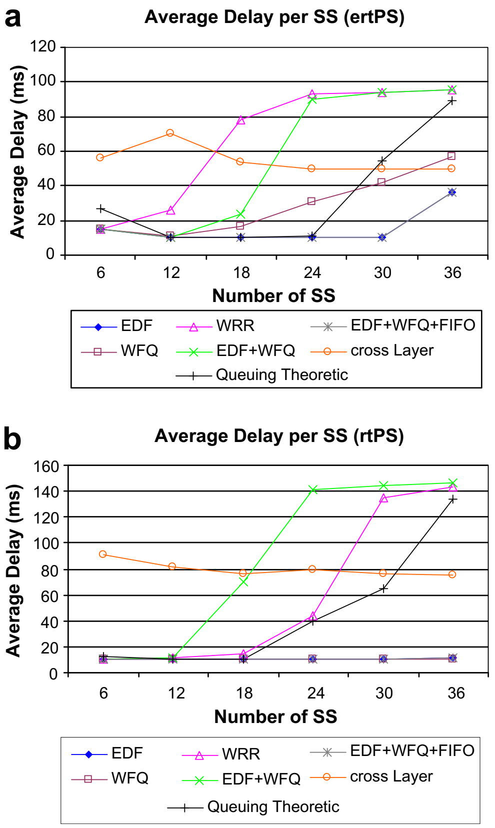

Fig. 3-4. The effect of uplink burst preamble – Average delay. (a) Average delay – ertPS. (b) Average delay – rtPS.

The Average delay and Packet loss increase with increasing number of SSs due to increasing overhead of uplink burst preamble and increasing number of SSs (Figs. 3-4 and 3-5). The WRR and hybrid (EDF + WFQ) algorithms show the highest Average delay with a large number of SSs. Since these algorithms partition bandwidth according to the MRTR and number of SSs, and due to their nature of selecting large number of SSs every frame, the amount of bandwidth allocated for SSs of ertPS and rtPS classes is less. Under the hybrid (EDF + WFQ) algorithm, both ertPS and rtPS SSs compete for bandwidth. This results in an increase in Average delay with increasing number of SSs. The Average delay experienced by the SSs under the Queuing Theoretic algorithm stays at 10 ms but rises sharply when number of SSs is large due to the high preamble overhead. The value of 10 ms corresponds to the allocation start time, which is the time when the uplink allocation begins. The Cross-Layer algorithm shows a relatively constant Average delay but a sharply rising Packet loss. The Packet loss under the Cross-Layer algorithm is very high as most of the packets miss their deadline since only one type of SS is scheduled most of the time.

3.2.3. The effect of frame length

According to the IEEE 802.16-2004 standard [1], the supported frame lengths for the WirelessMAN-OFDM PHY layer are 2.5, 4, 5, 8, 10, 12.5 and 20 ms. We studied the behavior of the algorithms with respect to frame utilization and the different frame lengths. Frame utilization is measured by dividing the total number of symbols used over the total symbols available in a frame.

Overall, the frame utilization peaks at approximately 78% due to the fixed number of SSs and the traffic load (Fig. 3-6). The frame utilization indicated by the WRR algorithm is lower due to the large packet size of video traffic (150–300 bytes) and FTP traffic (150 bytes). The large packet size together with the ability of the algorithm selecting all the SSs in a frame results in more symbols wasted. The Queuing Theoretic algorithm shows slightly higher

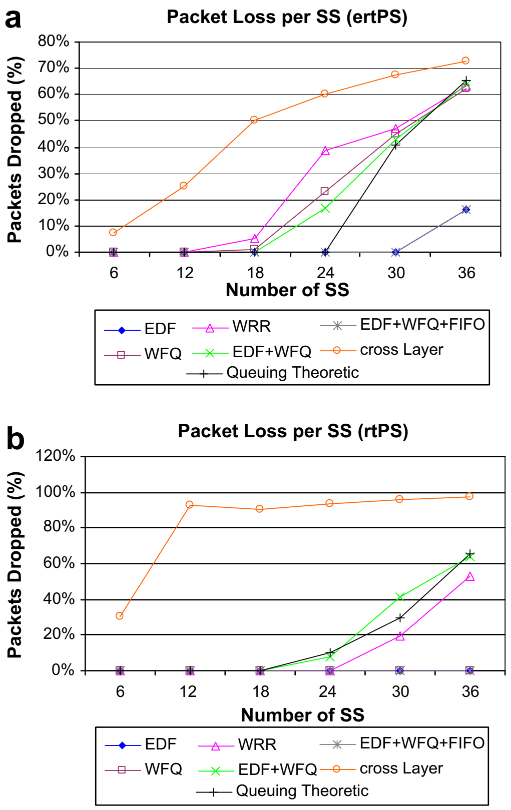

Fig. 3-5. The effect of uplink burst preamble – Packet loss. (a) Packet loss – ertPS. (b) Packet loss – rtPS.

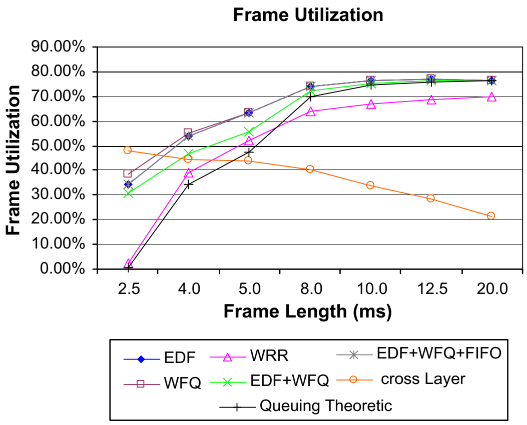

Fig. 3-6. The effect of frame length on frame utilization.

frame utilization than the WRR scheme, since the former algorithm is more efficient with variable packet sizes. With frame length smaller than 8 ms, the hybrid (EDF + WFQ) algorithm indicates lower frame utilization. As discussed previously, the frame utilization is lower due to the overall bandwidth partitioning of the hybrid (EDF + WFQ) algorithm that results in some unused bandwidth. Other legacy algorithm show similar performance with respect to frame utilization. The Cross-Layer algorithm shows the highest frame utilization when frame length is 2.5 ms but decreases sharply due to inefficient use of available bandwidth by the algorithm.

3.2.4. Bandwidth request analysis

In this experiment, we compare the piggyback and contention request mechanisms, as specified in the IEEE 802.16-2004 standard [1]. In the piggyback mechanism, a SS can attach its bandwidth request onto the data packets whereas in the contention mechanism a SS will compete for a slot to send its bandwidth request. Since piggyback requests do not have a type field, they will always be incremental requests. Due to the possibility of collisions, bandwidth requests under the contention mechanisms will always be aggregate requests.

We study the piggyback mechanism with the contention request mechanism under two different schemes of allocating slots for the contention mechanism. In the first scheme, a fixed number of slots, equal to the number of SSs, is reserved for contention. The second scheme is an enhancement to the first whereby a fixed number of slots, equal to half the number of SSs, are reserved and the SSs are also allowed to contend in the unused portion of the uplink sub-frame. The unused portion of the uplink sub-frame can be determined by each SS from the UL-MAP message. We have used nrtPS and BE SSs in 2:1 ratio i.e. for every two nrtPS SSs there is one BE SS present (Fig. 3-7).

We can observe that the contention mechanism is more suitable than the piggyback mechanism for the Cross-Layer algorithm (Fig. 3-8(a)–(c)). Due to the poor frame utilization of the Cross-Layer algorithm, a larger portion of the frame is available for contention. On the other hand, since the Cross-Layer algorithm selects only one SS in each frame, in most cases the selected SS does not have any backlogged packets to piggyback the bandwidth request thus resulting in very low overall bandwidth request success (honored requests).

The EDF algorithm results in the highest percentage of honored requests under the piggyback mechanism than all the other algorithms (Fig. 3-8(a) and (b)). This is due to both nrtPS and BE SSs competing for bandwidth that results in a large backlog of data and allows the SSs to piggyback bandwidth request. The hybrid (EDF + WFQ + FIFO) algorithm results in a lower percentage of honored requests under the piggyback mechanism than other legacy algorithms since it gives strict priority to nrtPS SSs. This will result in smaller backlog of data for nrtPS SSs and due to their high concentration (2:1 ratio), which translates into lower percentage of honored requests. The WFQ and hybrid (EDF + WFQ) algorithms also indicate a lower percentage of honored requests under the piggyback mechanism. This behavior is due to a higher weight assigned to nrtPS SSs that results in smaller backlog of data in their queues.

For fewer SSs (less than 12), the WRR algorithm shows a higher percentage of honored requests under the contention mechanism than under the piggyback mechanism. As a result of fewer SSs,

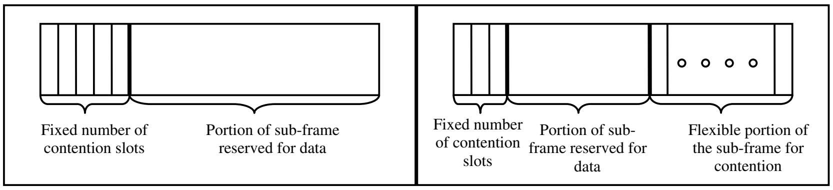
Fixed number of contention slots
Portion of sub-frame reserved for data
Fixed number of contention slots
Portion of sub-frame reserved for data
Flexible portion of the sub-frame for contention

Fig. 3-7. Uplink sub-frame structure: slot reservation for contention request mechanism. (a) Fixed number of contention slots. (b) Variable number of contention slots.

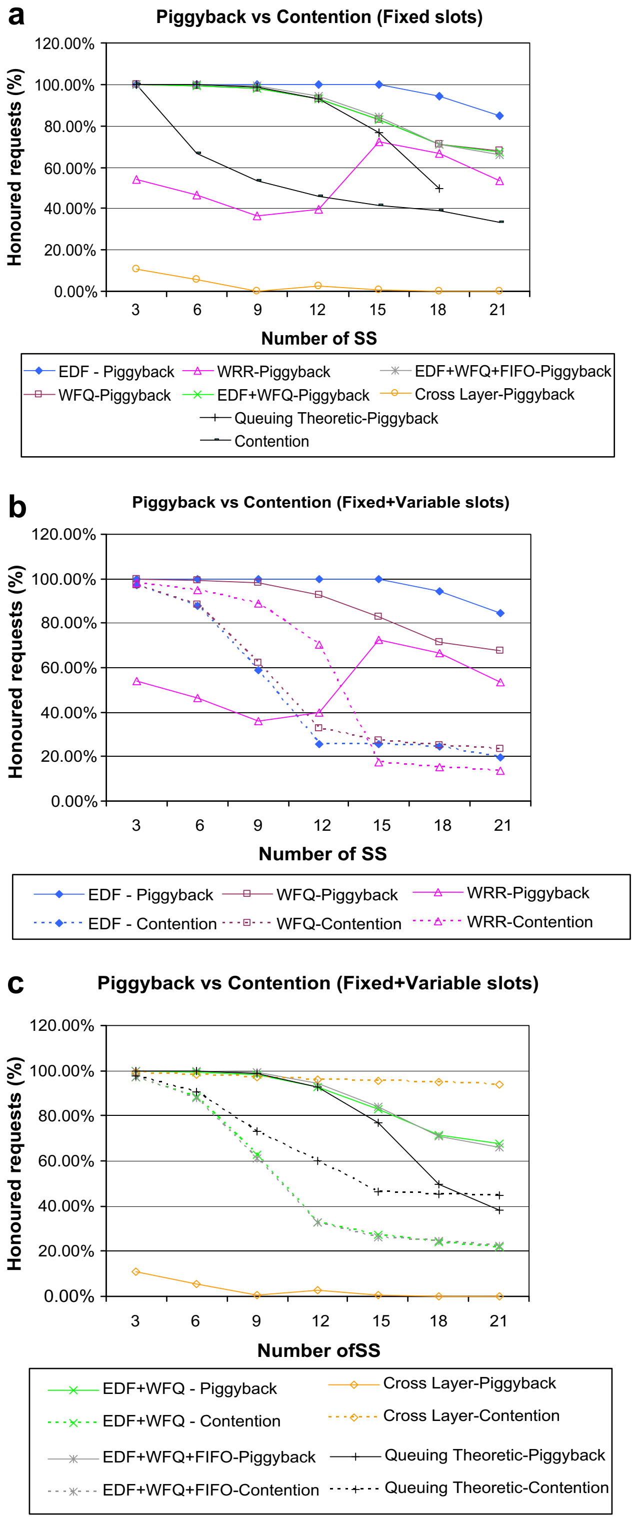

Fig. 3-8. Bandwidth request analysis. (a) Piggyback vs. contention (Fixed slots). (b) Piggyback vs. contention (Fixed + Variable slots) – homogenous schemes. (c) Piggyback vs. contention (Fixed + Variable slots) – hybrid and opportunistic schemes.

the bandwidth allocated to each SS is enough to flush out all the data resulting in fewer packets remaining to piggyback the bandwidth request. With a large number of SSs (greater than 12), the bandwidth allocated to each SS is not enough to transmit all the packets from the queue resulting in the piggyback mechanism performing better than the contention mechanism. When the number of SSs is greater than 15, the piggyback mechanism results in a decrease in honored requests. This behavior is caused by the lower bandwidth allocated to each SS and the large packet size of BE and nrtPS traffic. The small amount of bandwidth allocated is not enough to transmit one packet, and therefore a large number of SSs don't get selected. Even if a SS has large backlog of data, but if the SS is not selected in the current frame, it cannot piggyback on the bandwidth request. The Queuing Theoretic algorithm shows a higher percentage of honored requests under the contention mechanism (fixed + variable slots) than the legacy algorithms. This behavior is due to the limitation on the bandwidth allocation per class under the Queuing Theoretic algorithm. Due to a lower traffic arrival rate of BE SSs, some of the bandwidth allocated to the BE class will be unused resulting in more symbols available for contention.

4. Conclusions and recommendations

This paper presents a comprehensive performance evaluation of a number of algorithms for the uplink traffic in WiMAX networks. Existing proposals of scheduling schemes have been classified into three categories; homogenous, hybrid and opportunistic algorithms. Representative schemes from each of the categories have been evaluated with respect to major distinguishing characteristics of the WiMAX MAC layer as specified in the IEEE 802.16-2004 standard. Tables 4-1, 4-2 and 4-3 provide a summary of the results of our study.

Our study can be used to devise efficient scheduling algorithms for WiMAX PMP network that address some or all of the issues observed in the existing schemes. For instance, and based on the simulation results, we conclude that there is no single scheduling scheme that provides the desired performance with respect to all the QoS requirements and characteristics of the IEEE 802.16 MAC layer. A scheduling algorithm needs to be selected based on the requirements and traffic profiles of the network. Due to the emergence in popularity of real-time applications such as VoIP, audio/video streaming and video conferencing, many networks are dedicated for real-time traffic only. Such networks can be further classified into those consisting of a single type of application versus

Table 4-1
Comparison of homogeneous schemes

	EDF	WFQ	WRR
Intra-class fairness (ertPS/rtPS/nrtPS/BE)	High/high/high/high	Medium/high/medium/high	High/high/high/high
Inter-class fairness	Low	Medium	High
Frame utilization	High	High	Low–medium
Average Throughput (ertPS/rtPS/nrtPS/BE)	High/high/low/low	Medium/high/high/medium	Low/medium/medium/low
Average delay (ertPS/rtPS)	Low/low	Medium/low	High/high
Packet loss (ertPS/rtPS)	Low/low	High/low	High/medium

Table 4-2
Comparison of hybrid schemes

	EDF + WFQ	EDF + WFQ + FIFO
Intra-class fairness (ertPS/rtPS/nrtPS/BE)	High/high/medium/low–medium	High/high/medium/high
Inter-class fairness	Medium	Low
Frame utilization	Medium	High
Average Throughput (ertPS/rtPS/nrtPS/BE)	High/medium/high/low	High/high/medium/low
Average delay (ertPS/rtPS)	High/high	Low/low
Packet loss (ertPS/rtPS)	Medium/medium	Low/low

Table 4-3
Comparison of opportunistic schemes

	Queuing Theoretic	Cross-Layer
Intra-class fairness (ertPS/rtPS/nrtPS/BE)	Medium/low–medium/ low–medium/high	Low/low–medium/ low/high
Inter-class fairness	Medium	Low
Frame utilization	Low–medium	Low
Average Throughput (ertPS/rtPS/nrtPS/BE)	High/high/high/medium	Low/low/low/low
Average delay (ertPS/rtPS)	Medium/medium	Medium/medium
Packet loss (ertPS/rtPS)	Medium/medium	High/high

those with a mix of real-time applications. For the former type of network, EDF would be the most suitable as it shows low delay and Packet loss and high Average Throughput and intra-class fairness. Even though the EDF algorithm results in low inter-class fairness, since the networks would contain traffic from one class only, this should not be a concern. On the other hand, Queuing Theoretic scheme would be the most suitable for networks that consist of a variety of real-time applications. The Queuing Theoretic algorithm results in good performance for real-time applications at the same time providing reasonable fairness among the various real-time applications. The Queuing Theoretic algorithm would also be suitable for networks whose traffic profile predominantly consists of data traffic e.g. email, file transfer, and web browsing. A weakness of the Queuing Theoretic algorithm is that it does not include the Average Throughput in the utility function for rtPS SSs. According to the IEEE 802.16-2004 standard, MRTR is a mandatory QoS parameter for the rtPS class, and therefore a modification to the algorithm is necessary ensuring good performance of rtPS SSs.

If the main objective of deploying WiMAX networks is higher network bandwidth and at the same time reasonably satisfying the QoS requirements of all the traffic classes, WFQ would be a good choice. WFQ results in high frame utilization whilst maintaining medium to high Average Throughput for all the traffic classes. Normally, it would be expected that an opportunistic algorithm would be suitable for this purpose, since it exploits variations in channel quality and maximizes the throughput. The opportunistic algorithms evaluated in this study show low to medium frame utilization either due to their inability of selecting multiple SSs per frame or selecting maximum number of SSs per frame. Therefore, selecting maximum number of SSs in each frame does not result in the highest frame utilization due to the uplink burst preamble overhead.

The choice of bandwidth request mechanism significantly depends on the traffic load and the characteristics of the scheduling scheme. Except for the Cross-Layer algorithm, all algorithms show superior performance under both contention mechanisms when the traffic load is light. Under heavy load, the piggyback mechanism performs better due to larger backlog of data that allows the SSs to tag the bandwidth requests onto the data packets. The Cross-Layer algorithm exhibits a distinctive behavior in comparison to the other studied algorithms. This is because the Cross-Layer algorithm selects only 1 SS in each frame, it allows the selected SS to flush out all its backlogged packets thus none remain to piggyback the bandwidth request.

To address the issue of the Cross-Layer algorithm not utilizing the available bandwidth efficiently, we propose a modification to the algorithm that requires dividing the algorithm into two stages. In the first stage, sufficient bandwidth is allocated, in a Round Robin fashion, to satisfy the MRTR of the SSs. In the second stage, bandwidth is allocated according to the priority of the SSs, calculated as proposed in [10]. We have also observed from the study that WRR algorithm does not perform well when the traffic contains variable sized packets. Variably Weighted Round Robin (VWRR) [21], a variation of WRR, can be considered to address this shortcoming of the WRR scheme. The VWRR scheme is designed to adaptively change the weight of the SSs based on the mean packet size.

A Call Admission Control (CAC) scheme works in tandem with the scheduling algorithm in ensuring the number of connections in the network can be served reasonably well. Therefore, the choice of a CAC scheme is critical for the performance of a scheduling algorithm thus making it vital to evaluate the different scheduling schemes with respect to representative CAC algorithms. A significant component of the IEEE 802.16-2004 standard deals with the diverse set of bandwidth request mechanisms. Further investigation needs to be carried out to evaluate the scheduling algorithms under different bandwidth request mechanisms and CAC schemes.

References

[1] IEEE 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems, October 2004.

[2] IEEE 802.16e-2005, IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment. 2. Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, February 2006.

[3] C. Cicconetti, A. Erta, L. Lenzini, E. Mingozzi, Performance evaluation of the IEEE 802.16 MAC for QoS Support, IEEE Transactions on Mobile Computing (2007) 26–38.

[4] N. Ruangchaijatupon, L. Wang, Y. Ji, A study on the performance of scheduling schemes for broadband wireless access networks, in: Proceedings of the International Symposium on Communications and Information Technology, October 2006, pp. 1008–1012.

[5] D. Ferrari, D. Verma, A scheme for real-time channel establishment in wide-area networks, IEEE Journal on Selected Areas in Communications (1990) 368–379.

[6] T. Tsai, C. Jiang, C. Wang, CAC and Packet scheduling using token bucket for IEEE 802.16 networks, Journal of Communications 1 (2) (2006) 30–37.

[7] K. Wongthavarawat, A. Ganz, Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems, International Journal of Communication Systems 16 (1) (2003) 81–96.

[8] K. Vinay, N. Sreenivasulu, D. Jayaram, D. Das, Performance evaluation of end-to-end delay by hybrid scheduling algorithm for QoS in IEEE 802.16 network, in: Proceedings of the International Conference on Wireless and Optical Communication Networks, April 2006, 5 pp.

[9] M. Settembre, M. Puleri, S. Garritano, P. Testa, R. Albanese, M. Mancini, V. Lo Curto Performance analysis of an efficient packet-based IEEE 802.16 MAC supporting adaptive modulation and coding, in: International Symposium on Computer Networks, June 2006, pp. 11–16.

[10] Q. Liu, X. Wang, G. Giannakis, Cross-layer scheduler design with QoS support for wireless access networks, in: Proceedings of International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, August 2005, 8 pp.

[11] H. Rath, A. Bhorkar, V. Sharma, An opportunistic uplink scheduling scheme to achieve bandwidth fairness and delay for multiclass traffic in Wi-Max (IEEE 802.16) broadband wireless networks, in: Proceedings of IEEE Global Telecommunications Conference, November 2006, pp. 1–5.

[12] D. Niyato, E. Hossain, A queuing-theoretic optimization-based model for radio resource management in IEEE 802.16 broadband wireless networks, IEEE Transactions on Computers 55 (11) (2006) 1473–1488.

[13] V. Singh, V. Sharma, Efficient and fair scheduling of uplink and downlink in IEEE 802.16 OFDMA networks, in: Proceedings of IEEE Wireless Communications and Networking Conference, September 2006, pp. 984–990.

[14] S. Kim, I. Yeom, TCP-aware uplink scheduling for IEEE 802.16, IEEE Communications Letters (2007) 146–148.

[15] J. Xu, R. Lipton, On fundamental tradeoffs between delay bounds and computational complexity in packet scheduling algorithms, in: Proceedings of the 2002 SIGCOMM Conference, October 2002, pp. 279–292.

[16] H. Fattah, C. Leung, An overview of scheduling algorithms in wireless multimedia networks, Proceedings of IEEE Wireless Communications (2002) 76–83.

[17] K. Chen, A. Liu, L. Lee, A multiprocessor real-time process scheduling method, in: Proceedings of the Fifth International Symposium on Multimedia Software Engineering, December 2003, pp. 29–36.

[18] J. Chen, C. Wang, F. Tsai, C. Chang, S. Liu, J. Guo, W. Lien, J. Sum, C. Hung, The design and implementation of WiMAX module for ns-2 simulator, in: Proceedings of the 2006 Workshop on Ns-2: The IP Network Simulator, October 2006, 5 pp.

[19] B. Kim, Y. Hur, Application Traffic Model for WiMAX Simulation, POSDATA, Ltd., April 2007.

[20] M. Dianati, X. Shen, S. Naik, A new fairness index for radio resource allocation in wireless networks, in: Proceedings of IEEE Wireless Communications and Networking Conference, March 2005, pp. 712–717.

[21] Y. Ito, S. Tasaka, Y. Ishibashi, Variably weighted round robin queuing for core IP routers, in: Proceedings of Performance, Computing and Communications Conference, April 2002, pp. 159–166.