The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.108 e-ISSN: 2251-7545

Bandwidth Allocation Architecture for IEEE 802.16 QoS Scheduling

Ayesha Haider Ali
* LCWU
ayesha.iqbal@gmail.com

Faria Kanwal
LCWU
faria.kanwal@gmail.com

Afnan Bashir
COMSATS Institute of Information Technology,Lahore
afnan.bashir@hotmail.com

Kiran Zia
LCWU
ms_kz6@yahoo.com

Komal Bashir
LCWU
ko_junaid@yahoo.com

***Abstract*—Due to the growing dependence on web for information, entertainment, business, education and real time applications, the demand for high speed broadband wireless systems is increasing rapidly. The IEEE 802.16 (WiMAX) standard has emerged as a solution to meet all these requirements. The IEEE 802.16 has the following advantages; high data rate, wireless access for last mile, point to multipoint communication, high frequency range and QoS for various types of application flows. However the details of packet scheduling mechanisms are left unspecified in the standard. Therefore, we propose a QoS scheduling architecture for efficient bandwidth allocation. Our main goals are to provide delay and bandwidth guarantees for various type of applications and to maintain fairness among various flows.**

Keywords-WiMAX, QoS, IEEE802.16

I. INTRODUCTION

To improve the efficiency and competence of broadband wireless networks, great importance has been given to QoS. Keeping in view the strict QoS requirements of various applications the IEEE 802.16 standard has been developed but, it does not provide any details of scheduling packets belonging to different classes in order to meet their variedQoS requirements.[1][2] Therefore, an efficient QoS scheduling architecture is required in order to provide QoS guarantees to different types of applications.[3][4]

We have designed an efficient scheduling architecture for IEEE 802.16 with a fixed point to multipoint topology. Our main design objective is to provide delay and bandwidth guarantees to QoS sensitive applications and maintain fairness among various flows.

.

II. RELATED WORK

In [5], Weighted Round Robin (WRR) is suggested for uplink scheduling and the grant mode is GPSS. The authors have chosen to use chose five priority queues each being assigned dynamic priority. WFQ is used for higher priority services, WRR (Weighted Round Robin) for middle priority services and FIFO for lower priority services. The authors did no comment on what weights to use for WRR scheduling and BS downlink scheduling.

In [6] the authors suggest cross layer optimizations for enhancing the QoS of IEEE 802.16. Using DiffServ services, these optimizations perform traffic classifications and packet mapping. Mechanism for admission control is defined at the BS. A hierarchical scheduling algorithm is deployed at BS, which uses six queues in both uplink and downlink directions. Fixed bandwidth is allocated to UGS flows. Deficit Fair Priority Queue (DFPQ) is used for traffic scheduling. This work focuses more on cross layer QoS optimization rather than defining QoS scheduling algorithm for IEEE 802.16.

In [7], using GPC grant mode, the authors have suggested a scheduling architecture for IEEE 802.16 and DOCSIS (standard for delivering broadband services over Hybrid Fiber Coax). The work emphasizes more on DOCSIS rather than IEEE 802.16. Three types of queues are used namely: FIFO (for UGS flows and supports request for rtPS and nrtPS flows. FIFO queue(for flows with minimum reserved bandwidth) and priority queue(for flow with no bandwidth reservation). Prioritized WFQ (PWFQ) scheduling is employed for scheduling. However, they have not mentioned the BS downlink scheduling algorithm.

In [8], the authors have suggested allocation algorithms based on flow type and strict priority. Fixed bandwidth is allocated for UGS, Early Deadline Fisrt (EDF), WFQ is used for scheduling. A bandwidth allocation module for both uplink and downlink is proposed to avoid the usage of more bandwidth than allocated to the higher priority flows. The authors use simulation model to show the effectiveness of their algorithm.

In [11], the authors propose a Scalable QoS Scheduling Architecture (SQSA) for IEEE 802.16 to guarantee QoS demands of different applications, find the appropriate QoS that fulfils each flow requests and to ensure this QoS.

In [12], authors propose a cross-layer DiffServ architecture for wireless multi-hop mesh networks for end-to-end QoS support. Simulations results have been shown for the proposed QoS architecture using OPNET Modeler.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.108 e-ISSN: 2251-7545

III. PROPOSED ARCHITECTURE

Our proposed architecture implements GPSS mode for granting bandwidth in terms of time. The architecture supports all the four uplink services specified in IEEE 802.16.

The following design decisions are taken:

1. For granting bandwidth, GPSS mode has been selected due to the fact that it is scalable, efficient and allows SS to react faster to the real-time applications need.[9]
2. Based on their weight, BS allocates uplink bandwidth to each SS. More weight is assigned to the flows with higher priority to ensure that they are served before the lower priority flows.
3. WFQ algorithm is used by the SS to distribute allocated bandwidth among its flows based on their priority. [10]
4. WFQ is used to determine the order of packet transmission for different flows in downlink direction.

The proposed architecture is shown in Figure 1. Following is a brief overview of our architecture:

The following events take place at BS:

1. BS receives uplink request packets sent on uplink channel by different SSs. These packets are forwarded to the Bandwidth Allocator and the total bandwidth is divided among different flows and SS are then assigned to these flows.
2. BS then schedules data packets and creates downlink control message.
3. BS transmits downlink data for various SSs.

The following events take place at the SS:

1. The received packet is classified and placed in one of the uplink queues.
2. SS also issues bandwidth requests that are to be sent to the BS.
3. The uplink data packets are then scheduled by SS in the allocated slot and are transmitted accordingly.
4. After finishing the uplink transmission, SS once more starts listening on the downlink channel.

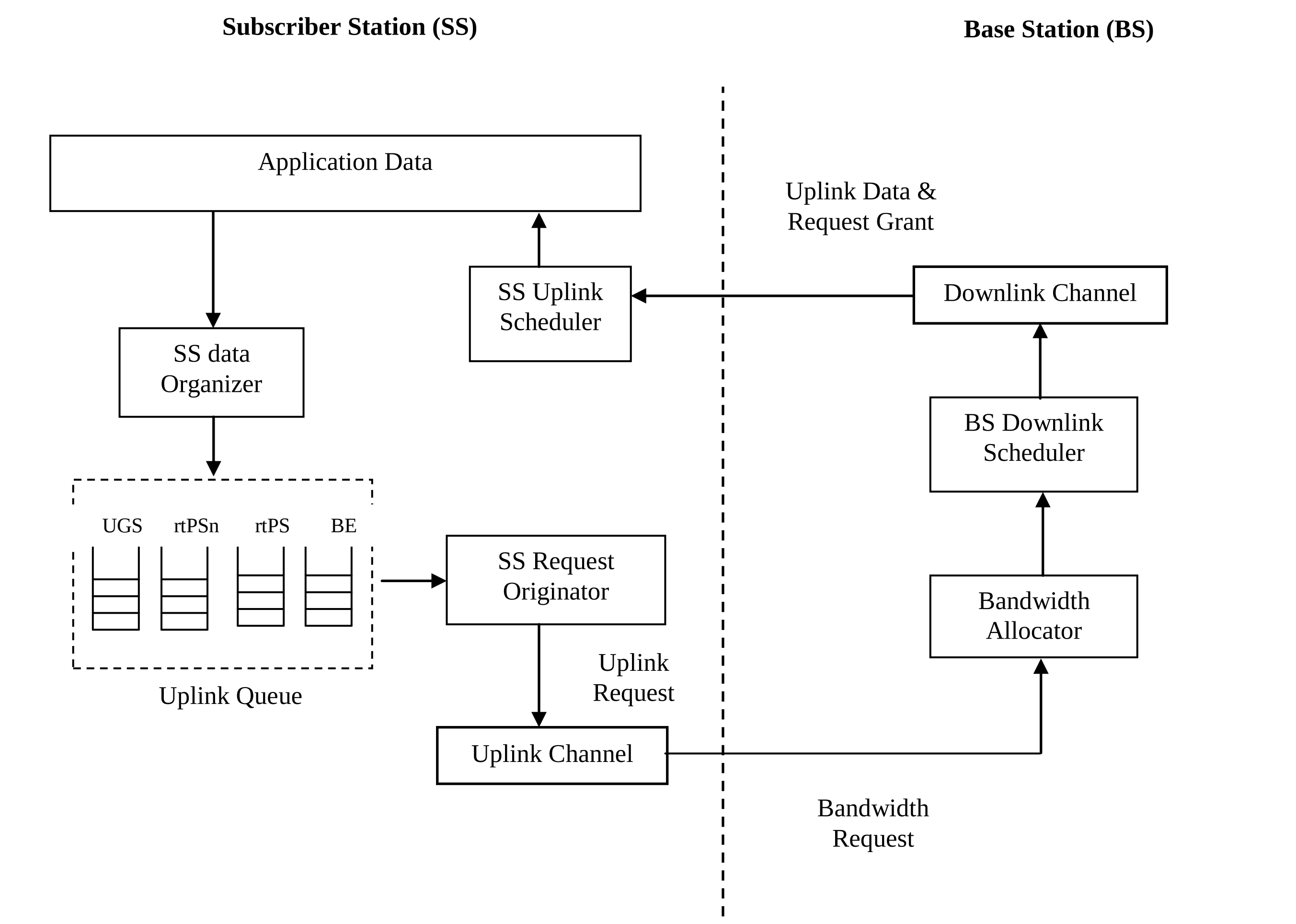

Figure1: Proposed Architecture

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.108 e-ISSN: 2251-7545

A. Components of SS

SS Data Organizer

This component is responsible for classifying the incoming data into uplink queues based on the type of traffic. There are four queues for each of the four uplink service classes. UGS traffic is mapped on to UGS queue, rtPS traffic is mapped on to rtPS queue, nrtPS traffic is mapped on to nrtPS queue and BE traffic is mapped on to BE queue.

SS Request Originator

The bandwidth request generated by various connections is handled by *SS Request Originator*. For each connection aggregate request is generated for all of its flows

SS Uplink Scheduler

The scheduling of uplink traffic is done by the SS Uplink Scheduler. It divides the entire bandwidth of the uplink among active SS according to the bandwidth) it has received and the QoS attributes of each connection of these SS.

To serve various uplink flows, WFQ scheduling is used to schedule the traffic. UGS traffic is given the highest priority. Next the packets of Type rtPS and nrtPS are served.

After serving all the high priority traffic flows, BE is served with remaining bandwidth. FIFO is used to serve the queues of same type.

B. Coponents of BS

Bandwidth Allocator

This component is responsible for allocating bandwidth to each SS. Amount of bandwidth allocated depends on the following factor:

- Bandwidth requested by each SS for transmitting data in uplink direction.

Assigning flow bandwidth to SS

The bandwidth allocated to different flowsis distributed among all SS. Bandwidth allocated to UGS flow is equally distributed among all SS. After that bandwidth allocated to all other flows is distributed in order of their priority.

The order of transmission among SSs is decided according to the deadline of UGS data to be transmitted by each SS, same is the case for all other flows.

The following Algorithm is used to assign flow bandwidth to SS. Following notations are used in the algorithm

TotalBytes = Total bytes available for uplink transmission
SS = Total number of active SS
RequestUGS[i] = Bytes requested by ith SS for UGS flow
RequestnrtPS[i] =Bytes requested by ith SS for nrtPS flow
RequestrtPS[i] = Bytes requested by ith SS for rtPS flow
RequestBE[i] = Bytes requested by ith SS for BE flow
WeightUGS = 4
WeightnrtPS =3
WeightrtPS = 2
WeightBE = 1
Req[i] =Total bytes requested for ith flow by all SS
Alloc[i] =Total bytes allocated to ith flow for all SS
Weight[i] = Weight assigned to ith uplink flow
RequestSS[i][j] = Total bytes requested for ith flow by jth SS
flowAllocSS= Total bytes allocated to ith SS
SSBytes = Total bytes granted to ith SS
satisfied[j] =Keeps track of satisfied SS

Algorithm for allocating bandwidth to SS

Req[1]= Req[2]= Req[3]= Req[4]=0
Alloc[1]= Alloc[2]= Alloc[3]= Alloc[4]=0
Weight[1]=WeightUGS
Weight[2]=WeightrtPS
Weight[3]=WeightnrtPS
Weight[4]=WeightBE
SS= N
for j=1; j<=SS;j++
RequestSS[1][j] = RequestUGS[j]
RequestSS[2][j] = RequestrtPS[j]
RequestSS[3][j] = RequestnrtPS[j]
RequestSS[4][j] = RequestBE[j]
Req[1] + = RsequestUGS[j]
Req[2] + = RequestrtPS[j]
Req[3] + = RequestnrtPS[j]
Req[4] + = RequestBE[j]
end for

The following part of the algorithm assigns flow bandwidth to SS

whileAlloc[i] > 0 do
flowAllocSS = Alloc[i] / SS
if satisfied[j] == 0 then
for j=1;j<=SS; j++
if flowAllocSS>= ReqSS[i][j] then //total bytes granted to jth SS for all the flows
SSBytes[j] += ReqSS[i][j]
ReqSS[i][j] =0
satisfied[j] =1
else
SSBytes [j] += flowAllocSS
ReqSS[i][j] -= flowAllocSS
end if
end if
end for
end while

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.108 e-ISSN: 2251-7545

BS Downlink Scheduler

The scheduling of downlink traffic is done by the *BS Downlink Scheduler*. It distributes the entire downlink bandwidth among downlink connections according to its priority. Packets from UGS flows are transmitted first. Remaining bandwidth, if any, is then distributed to rtPS flows and so on. WFQ scheduling is used to schedule the traffic.

IV. EXPERIMENTS AND RESULTS

The simulation setup consists of one BS and a number of SS as shown in Figure 2.

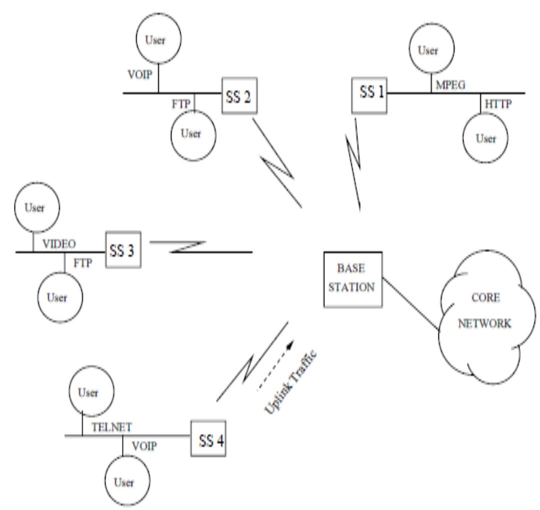

Figure 2: Simulation Setup

Following table shows the simulation parameters.

Table 1: Simulation Parameters

Parameters	Values
No. of BS	1
No. of SS	Multiple
MAC layer	802.16
Physical layer	802.11b
Bandwidth	11 Mbps
Frame duration	10 ms
Application flows	VOIP, FTP, TELNET
Performance metrics	Average delay of uplink flows at SS, effective bandwidth utilization

We have assumed that number of SS cannot change dynamically during the course of simulation. The application flows shown above are mapped onto the appropriate uplink scheduling flows.

The following experiment shows average delay comparison of various uplink flows for each SS.

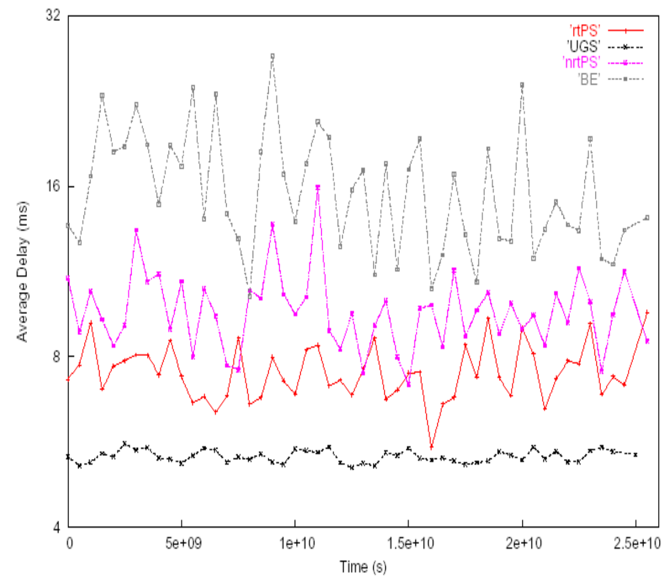

Figure 2: Comparison of Average Delay and Time(Load offered by the higher priority traffic is more than the lower priority traffic)

The following figure shows the results of the above experiment when the load offered by the lower priority traffic is higher than the high priority traffic.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 1-2, 2013
Doi: 10.7321/jscse.v3.n3.108 e-ISSN: 2251-7545

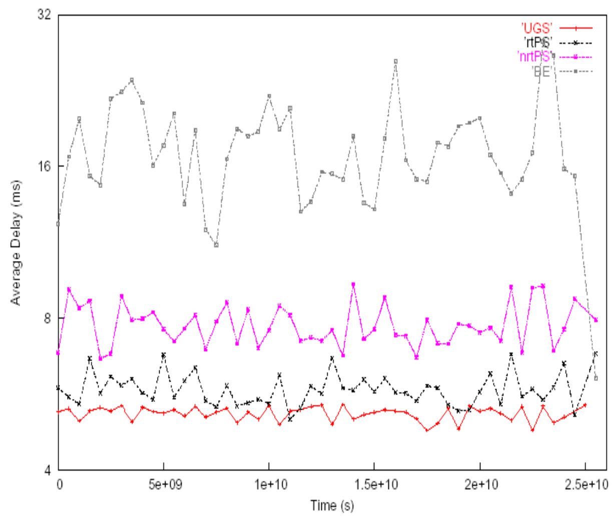

Figure 3Comparison of Average Delay and Time (load offered by the lower priority traffic is higher than the high priority traffic)

The next experiment shows the effect of the number of SS on delay

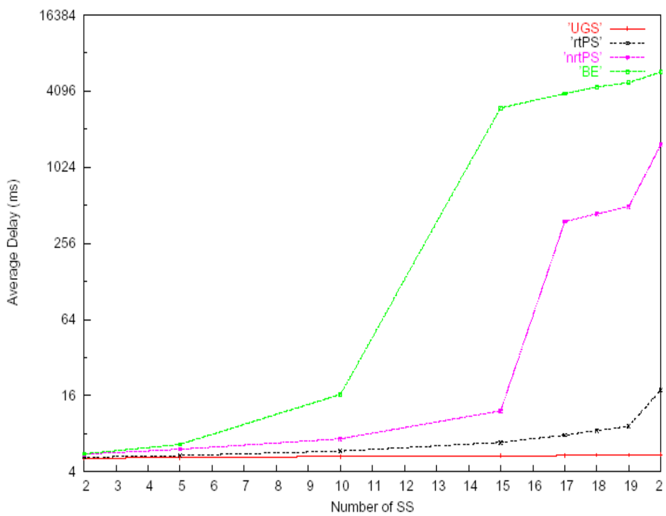

Figure 4: Delay and Number of SS

The following figure shows the relationship between number of SS and average delay for all the four flows.

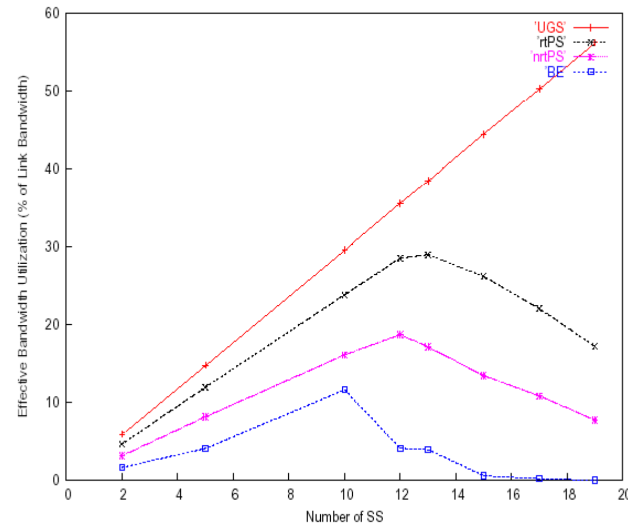

Figure 5: Number of SS and its effect on bandwidth Utilization (for all flows)

Bandwidth utilization for lower priority flows decreases as the system load increases.

V. CONCLUSION

A QoS scheduling architecture for IEEE 802.16 has been proposed which ensures efficient bandwidth allocation. The main purpose of our work is to ensure QoS guarantees to various traffic flows and applications and to allocate bandwidth in such a way so as to reduce delays and gain high throughput. It supports the QoS requirements of all four types of service flows mentioned in the standard. Priority of different type of traffic is considered to maintain efficient and fair resource/ bandwidth allocation.

REFERENCES

[1] C. Eklund, R.B. Marks, K. I. Stanwood and S. Wang: IEEE Standard 802.16; A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access, IEEE Communication Magazine: June 2002
[2] Air Interface for Fixed Broadband Wireless Access Systems", IEEE Standard for Local and Metropolitan Area Networks Part 16: Oct 2004
[3] Hamed S. Alavi, Mona Mojdeh and Nasser Yazdani: A Quality of Service Architecture for IEEE 802.16 Standards, Asia-Pacific Conference on Communications: 2005
[4] Claudio Cicconetti, Luciano Lenzini, and EnzoMingozzi: Quality of Service Support in IEEE 802.16 Networks,IEEE Network, March/April 2006
[5] Guosong Chu, Deng Wang, and Shunliang Mei: A QoS architecture for the MAC Protocol of IEEE 802.16 BWA System, IEEE International Conference on Communications Circuits and System and West Sino Expositions: 2002
[6] Jianfeng Chen, Wenhua Jiao, QianGuo: Providing integrated QoS control for IEEE 802.16 broadband wireless access systems, Vehicular Technology Conference 2005
[7] M. Hawa and D. W. Petr: Quality of service scheduling in cable and broadband wireless access systems, 10th IEEE International Workshop on Quality of Service: May 2002

Doi: 10.7321/jscse.v3.n3.108 e-ISSN: 2251-7545

[8] Aura Ganz and KittiWongthavarawat: IEEE 802.16 based last mile broadband wireless military networks with quality of service support, IEEE Milcom 2003
[9] L. F. M. de Moraes, P. D. Maciel Jr.: Analysis and Evaluation of a New MAC Protocol for Broadband Wireless Access, International Conference on Wireless Networks, Communications, and Mobile Computing – WirelessCom: June 2005.
[10] Weibin Zhao, David Olshefski and Henning Schulzrinne: Internet Quality of Service, Technical Report CUCS-003-00, Department of Computer ScienceColumbia University, 2000
[11] Bayan, A.F., Tat-Chee Wan, A scalable QoS scheduling architecture for WiMAX multi-hop relay networks, Education Technology and Computer (ICETC), 2010 2nd International Conference, 22-24 June 2010