Volume 2, Issue 7, July 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Queue-Length Based Approach for Scheduling In IEEE 802.16 Networks

Akashdeep[1], Dr. K. S. Kahlon[2]
[1]Department of Computer Science Engineering, UIET Panjab University Chandigarh.
[2]Department of Computer Science Engineering, GNDU Amritsar

Abstract: The popularity of WiMAX and the fact that problem of scheduling has been left as open issue by IEEE, it has become one of the hot areas of research. In this paper we had proposed a technique based on calculating the length of the queue. The proposed scheduler provides fair opportunities to different kinds of traffic. Results demonstrate the effectiveness of our algorithm.

Keywords: WiMAX, QoS, Scheduling

I. Introduction

Wireless and mobile communications have changed the communication systems over the past decades. The staggering growth of Internet has added additional demand for high speed wireless internet access, voice and multimedia applications. IEEE 802.16 also known as WiMAX (Worldwide Interoperability for Microwave access Networks) has been designed to provide wireless and wired broadband access with QoS guarantees in Metropolitan area networks [1][2]. From the initial variants, the IEEE 802.16 standard has undergone several amendments and evolved to the IEEE 802.16.2009 which has been amended by IEEE 80216j, 802.16h and 802.16m is the currently active standard.

WiMAX operates in two modes, compulsory Point to Multi Point (PMP) mode and an optional Mesh mode. The problem of scheduling differs in both the modes. In Mesh mode the scheduling is either central in which the mesh BS schedules all SSs or distributed where each station in two hop neighborhood must have their transmissions coordinated to avoid collisions. A number of schedulers have been proposed based on queuing theory. The simplest scheduling algorithm employed for scheduling of IEEE 802.16 networks is the Round Robin (RR) scheduler that was used to nullify the decision time required to be taken to schedule every packet. This type of scheduler works simply on a round robin basis by distributing equal channel resources to all the SSs without any priority. However, this technique is not suitable for systems with different levels of priority and systems with strongly varying sizes of traffic [6] as it does not support priority. There is an extension of the RR scheduler, the Weighted Round Robin (WRR) scheduler that is based on static weights and in which weights are assigned to every flow/queue and scheduling takes place in accordance with these weights [8]. Qualnet simulator from scalable network technologies had this scheduler as default in its implementation of IEEE 802.16 networks [7]. The drawback of WRR is that it is difficult to implement and Sayenko et al. [8] insisted that WRR is not fit for IEEE 802.16 networks as weights are floating numbers while slots allotted are integers. M Shreedhar et al.[9] has implemented DRR scheduler that associates a fixed quantum and a deficit counter with each flow/queue i. At the start of each round and for each flow i, deficit counter is incremented by fixed quantum. The head of the queue[i] is dequeued if deficit counter is greater than the length of the packet waiting to be sent. In this case, counter is decremented by length of packet. At each round, one packet at most can be sent for each flow. H. K. Rath et al [10] had proposed a new variant of DRR scheduler namely Opportunistic Deficit Round Robin (O-DRR) that handles latency critical applications. Reference of few more scheduling algorithms like Proportional Fairness, Drop tail queue, random early detection and random early detection with IN/OUT may be found in [11][12][13] in which comparisons between various scheduling techniques have been carried out by respective authors.

II. Materials and Methods

MAC protocol of IEEE 802.16 is connection oriented to support multiple QoS classes for accommodating heterogeneous traffic. According to the standard, the BS scheduler provides grants (time slots) at periodic intervals to the UGS flows to send data. Periodic grants are also given to rtPS and nrtPS flows to request bandwidth. Before satisfying bandwidth requests, the uplink scheduler must allocate resources to these periodic grants. The scheduler shall guarantee that delay and bandwidth requirements of rtPS and nrtPS flows are always satisfied. Each SS sends a request message containing its class and traffic specifications, this request

© 2012, IJARCSSE All Rights Reserved

can be sent as multicast, broadcast or piggy back request message. BS accepts the connection when enough resources are available to accommodate new connection as well as to serve the existing connections and performs the bandwidth allocation according to an appropriate scheduling scheme.

The authors of [14] have used three different queues to schedule different kinds of traffic namely low, high and intermediate priority queues. High priority queue consist of periodic grants for UGS flows and unicast request opportunities for rtPS and nrtPS flows, it includes flows that must be scheduled in the next frame. Intermediate queue is used for holding bandwidth requests that have been sent by rtPS and nrtPS connections while low priority queue handles bandwidth requests for BE traffic class. The priority for queues has been assigned in accordance with IEEE standard 802.16. The queues were served using strict priority ie high priority followed by medium and low priority however improvement was made as the requests whose deadline is going to expire can migrate to high priority queue. A method to calculate deadlines was also specified in which arrival time of the packet at SS is considered as worst case since BS does not have this information and is taken as equal to the sum of the arriving time of the last request sent by the connection and its maximum delay requirement. The drawback of this method is that it adds a significant overhead to the system as intermediate and low priority queue need to be analyzed every time scheduler is executed to check for request migration and secondly it can be unfair to flows as outlined below.

The processes in intermediate priority queues can migrate to high priority queues on the basis of their deadlines only however the method used for migration can be inefficient as there was no provision to migrate packets on the basis of congestion or length of queues. It is possible that high priority queue may be empty at some point of time and low and intermediate queues might be flooded with number of requests that may eventually lead to packet drops. Since only those requests can migrate whose deadline is going to expire in the next frame therefore packets in low and intermediate queues whose deadline is not expiring will not be able to migrate even though high priority queue remains idle. Therefore the technique presented by the authors may be in efficient and will be unfair to BE flows that may be present in low priority queue.

The technique presented in this paper tries to improve the shortcomings of the previous technique as the process of migration will not only be based on the deadline but also on the length of queues. The length of the queues will be monitored after a fixed span of time (Δt) and if the length is greater than specified threshold, the algorithm performs the migration of the packets from that particular queue to high priority queue. Also packets from low priority queues can be allowed to migrate to high priority queue. The number of packets to be migrated from intermediate queue depends on the arrival time of the packets, the packets will be sorted on the basis of arrival time and those packets will be migrated whose deadline is going to expire within (Δt) time. The reason for using this approach is to limit number of packets to be migrated to high priority queue as it will not cause bottleneck at high priority queue. The algorithm can be stated as follows:-

ALGORITHM

1. insert in the high priority queue, the periodic data grants and unicast request opportunities that must be scheduled in next frame.
2. Initialize variables Δt , $\mu 1$ to suitable values
3. For each Δt time intervals do
4. calculate queulength of intermediate queues;
5. if length_inter[queue]>$\mu 1$
6. CheckDeadline
7. CheckMinimumBandwidth;
8. schedule the requests in the high priority queue starting from the head of the queue;
9. if intermediate queue is empty and available slots > 0 then schedule the requests in the low priority queue starting from the head of the queue;

CheckDeadline:

10. for each request *i* in the intermediate queue do
11. if service[CID] = = rtPS then
12. frame[i] = (*deadline*[*i*] − *current time*) ÷ *frame duration*;
13. if frame[i] = = 1 then
14. if available bytes ≥ BR[i]
15. migrate request *i* to high priority queue;
16. granted BW[CID] = granted BW[CID] + BR[i];
17. backlogged[CID] = backlogged[CID] − BR[i];
18. available bytes = available bytes − BR[i];
19. if frame[i]>1 and frame[i]< Δt
20. if (available bytes/2) ≥ BR[i];
21. migrate request *i* to high priority queue;
22. granted BW[CID] = granted BW[CID] + BR[i];
23. backlogged[CID] = backlogged[CID] − BR[i];
24. available bytes = available bytes − BR[i];

***CheckMinimumBandwidth*:**

25. For each connection CID of type rtPS or nrtPS do
26. backlogged tmp[CID] = backlogged[CID];
27. granted BW tmp[CID] = granted BW[CID];
28. for each request *i* in the intermediate queue do
29. if BWmin[CID] ≤ granted BW tmp[CID] then
30. priority[i] = 0;
31. else
32. priority[i] = backlogged tmp[CID] −(granted BW tmp[CID] − BWmin[CID]);
33. granted BW tmp[CID] = granted BW tmp[CID] + BR[i];

© 2012, IJARCSSE All Rights Reserved

34. backlogged tmp[CID] = backlogged tmp[CID] −BR[i];
35. sort the intermediate queue;
36. For each request *i* in the intermediate queue do
37. if available bytes ≥ BR[i] then
38. migrate request to the high priority queue;
39. granted BW[CID] = granted BW[CID] + BR[i];
40. backlogged[CID] = backlogged[CID] − BR[i];
41. available bytes = available bytes − BR[i];

The above mentioned algorithm uses the same process to calculate priority as defined by [14] the change in algorithm is at the start where we calculate the length of queues for every Δt time periods, line number 3 and if the length is found to be greater than a pre-specified threshold, procedure *checkdeadline* is called, line 7; Our procedure to migrate the request is bit different as it first allows those requests to migrate whose deadline is going to expire in the next frame. If there is substantial bandwidth left then it performs the migration of those requests whose deadline is expiring in next Δt units. However this migrations has been limited as it may eventually drain out the whole bandwidth and may result in allocation of all the bandwidth to all requests in intermediate queue, in our case this has been handled at line no 20. The next section demonstrates the effectiveness of our method with help of simulations.

III. Results and Discussions

The topology of the simulated network consisted of a BS wire-attached to a fixed node through a 100 Mbps link with a 2 ms delay. The frame duration was 5 ms and the capacity of the channel was 40 Mbps, assuming a 1:1 downlink-to-uplink TDD split. Each SS is assumed to have only one type of traffic flow: voice, video, FTP and WEB, which are associated to UGS, rtPS, nrtPS, and BE services. Simulation parameters are specified in Table I.

Table I: Network Simulation Parameters

Traffic Flow	Traffic Class	Source	QoS requirements
Voice	UGS	On/off mode with mean of 1.2 and 1.8sec for on/off period	UGS Grant Interval = 20ms
Video	rtPS	MPEG traces	Unicast request interval=20ms Delay=100ms
FTP	nrTPS	Exponential distribution with mean of 512Kbytes	Interval=1s;min bandwidth = 200kbps
WEB	BE	Normal distribution with area .88 and mean 7247bytes	----------------

Experiment 1

The aim of this experiment is to investigate the effect of increase in the number of SS. There are 10 SS and one BS and the number of nrtPS connections were varied from 10 to 35. The graph of fig 1 shows that scheduler provides fair opportunities to all SS.

Experiment 2

This experiment verifies the impact of the load increase of the UGS traffic service on the performance of other service classes. The scenario for this experiment consists of one BS and 50 SSs. There are 20 nrtPS connections, 20 BE connections and the number of active UGS connections varies from 1 to 20. The graph shows that there is not much effect on throughput of nrtPs traffic however performance of BE traffic tends to decrease as high priority traffic increases.

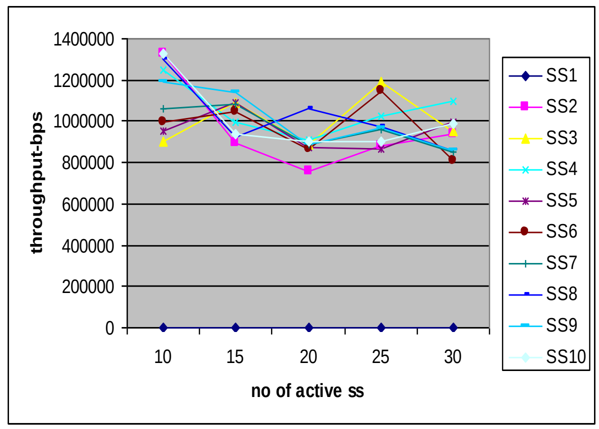

Fig. 1 Throughput of each SS

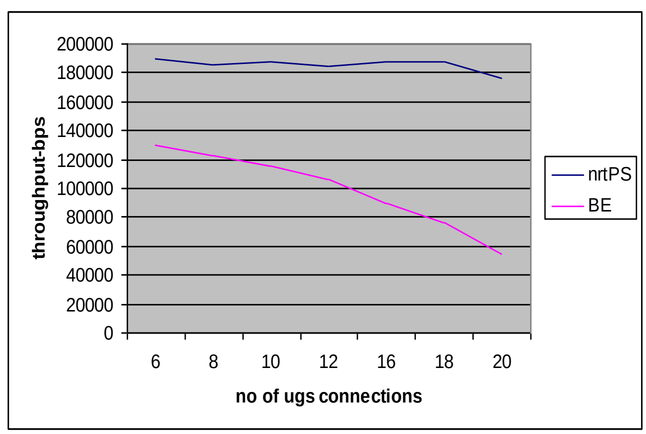

Fig. 2 Throughput of BE and nrtPS traffic

IV. Conclusion and Future Work

The paper discussed a novel approach for scheduling of IEEE 802.16 networks based on calculation of queue length. The algorithm checks for queue length after a fixed time interval and performs the migration of requests from one queue to another. The results taken are quite promising and are successful in providing the desired level of QoS. Future directions of research can be

© 2012, IJARCSSE All Rights Reserved

the application of the same approach to mesh mode and the process of migration can be based on cross layer parameters. A suitable optimization technique can be used to automate the process of migration.

References

1. IEEE Std. 802.16-2004, "IEEE standard for local and metropolitan area networks- part 16: Air interface for fixed broadband wireless access systems," 2004.
2. IEEE Std. 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 and corrigendum 1" 2005.
3. Ikbal Chammakhi Msadaa, Daniel Câmara, and Fethi Filali, "Scheduling and CAC in IEEE 802.16 Fixed BWNs: A Comprehensive Survey and Taxonomy", *IEEE Communications survey and Tutorials*, vol 12 no 4, Fourth Quarter 2010.
4. Miray Kas, Burcu Yargicoglu, Ibrahim Korpeoglu, and Ezhan Karasan, "A Survey on Scheduling in IEEE 802.16 Mesh Mode",*IEEE Communication Surveys & Tutorials*, Vol. 12, No. 2, Second Quarter 2010 pp 205-221.
5. Cicconetti, "Performance evaluation of the IEEE 802.16 MAC for QoS support", *IEEE Transaction on Mobile Computing*, vol 6, no 1, 2007 pp 26-38.
6. V. Sagar, D. Das, "Modified EDF algorithm and WiMAX architecture to ensure end to end delay in multihop network", *IEEE Region 10 Conference, Hyderabad*, pp 1-6, 19-21 November 2008.
7. www.scalable-networks.com
8. A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," in *Proc. 9th ACM International Symp. Modeling Analysis Simulation Wireless Mobile Syst.*. Terromolinos, Spain: ACM Press New York, NY, USA, 2006, pp. 108–117.
9. M. Shreedhar and G. Varghese, "Efficient fair queueing using Deficit Round-Robin," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 375-385, Jun. 1996.
10. H. K. Rath, A. Bhorkar, and V. Sharma, "An Opportunistic DRR (O-DRR) Uplink Scheduling Scheme for IEEE 802.16-based Broadband Wireless Networks", *IETE, International Conference on Next Generation Networks(ICNGN), Mumbai*, 9 February 2006
11. Loutfi Nuaymi and Aymen Belghith, "Comparison of WiMAX scheduling algorithms and proposals for the rtps QoS class", *14th European Wireless Conference, Prague*,pp1-6,22-25 June 2008.
12. Jani Lakkakorpi, Alexander Saenko, and Jani Moilanen, "Comparison of Different Scheduling Algorithms for WiMAX Base Station: deficit Round-Robin vs. Proportional Fair vs. Weighted Deficit Round-Robin", *IEEE conference on wireless communications and Networking,LasVegas*, pp 1991-1996, March 31 2008- April 3 2008.
13. Jin-Cherng Lin, Chun-Lun Chou, Cheng-Hsiung Liu "Performance Evaluation For Scheduling Algorithms In WiMAX Network " *22nd International Conference on Advanced Information Networking and Applications*, pp 68-74, 2008.
14. Juliana Freitag, Nelson L.S. da Fonseca, "Uplink Scheduling with Quality of Service in IEEE 802.16 Networks" *in IEEE Global Telecommunications Conference, GLOBECOM 2007*, 26-30 Nov 2007, pp 2503 -2508.

© 2012, IJARCSSE All Rights Reserved