Cross-Layer Routing-Scheduling in IEEE 802.16 Mesh Networks

Mehmet S. Kuran Gürkan Gür Tuna Tugcu Fatih Alagöz
Department of Computer Engineering
Bogazici University
Bebek 34342, Istanbul, TURKEY
{sukru.kuran, gurgurka, tugcu, alagoz}@boun.edu.tr

ABSTRACT

Broadband wireless access networks will be a vital component in IP-based fourth-generation (4G) wireless communication systems as part of convergent and pervasive networking architecture. IEEE 802.16 Mobile WiMAX is currently one of the most active standards for broadband wireless access. There are various challenges for the integration of WiMAX and next-generation broadband networks such as diverse operational environment, increasingly stringent QoS support, power/coverage limitations and capacity boundaries. The mesh operation mode of IEEE 802.16 has been standardized as a potential remedy to alleviate these issues. In this work, we propose and investigate a cross-layer routing-scheduling scheme in IEEE 802.16 mesh networks. Our scheme jointly utilizes the distributed and centralized scheduling capabilities of IEEE 802.16 link layer in mesh mode and routing in network layer for optimal operation. It abides with the rules of the IEEE 802.16 protocol. The experimental results indicate that our technique can significantly improve the network performance especially in case of a congestion in the Internet part of the traffic at the cost of a minor burden on the intranet traffic in the form of a slight increase in the end-to-end delay.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*network communications, wireless communication*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*routing protocols*

General Terms

Algorithms, Performance, Design

Keywords

IEEE 802.16, Wireless mesh networks, Cross-layer design, Routing, Scheduling

Mobilware'08, February 12-15, 2008, Innsbruck, Austria.

1. INTRODUCTION

In recent years, the proliferation of rich, mobile and personalized multimedia applications over Internet has increased the demand for broadband wireless access dramatically. This has led to the standardization of numerous wireless access standards such as IEEE 802.11 family, WiBro, IEEE 802.16 including Mobile WiMAX, ETSI HiperMAN and ETSI HiperACCESS. These wireless broadband access technologies are explained in detail in [1]. Mobile WiMAX is currently one of the most active standards for broadband wireless access. Broadband access networks based on WiMAX will provide crucial capabilities such as near-ubiquitous coverage, support for mission-critical applications in a flexible and standards-based manner.

WiMAX has two operation modes: *Point to Multipoint* (*PMP*) and Mesh. All users must be connected to the *Base Station* (*BS*) in the PMP mode. On the other hand, in the Mesh mode user stations that cannot connect directly to the BS can relay their signals over other subscriber stations (SSs) in range. Also, in the Mesh mode users can communicate directly with each other. The enabling of intranet traffic leading to larger capacity and increased coverage makes the Mesh mode of WiMAX a more versatile system than the PMP mode.

Being a broadband access technology most of the traffic generated by a WiMAX user is destined to the Internet. However, some network capacity is allocated also for the intranet traffic. Thus, in the standard the total system capacity is divided into two parts, one for Internet traffic and the other for intranet traffic. While the Internet traffic is managed by the BS, intranet traffic is managed by the *SSs*. This quasistatic allocation scheme has one major drawback: In cases where offered Internet traffic exceeds the allocated capacity congestion occurs even if the intranet traffic part is low. Since the BS does not know how much of the intranet part is being used at a given time, it cannot take any action to increase the capacity allocated to the Internet traffic.

WiMAX standard is a data link layer protocol. Thus, it does not specify how the traffic will be routed in the Mesh topology. In this paper we develop and evaluate a cross-layer *Centralized Queue Aware Routing* (*CQAR*) mechanism that uses the queue length values of WiMAX links as decision parameters and changes the routing of Internet traffic in case of congestion to reduce Internet traffic's ETE packet

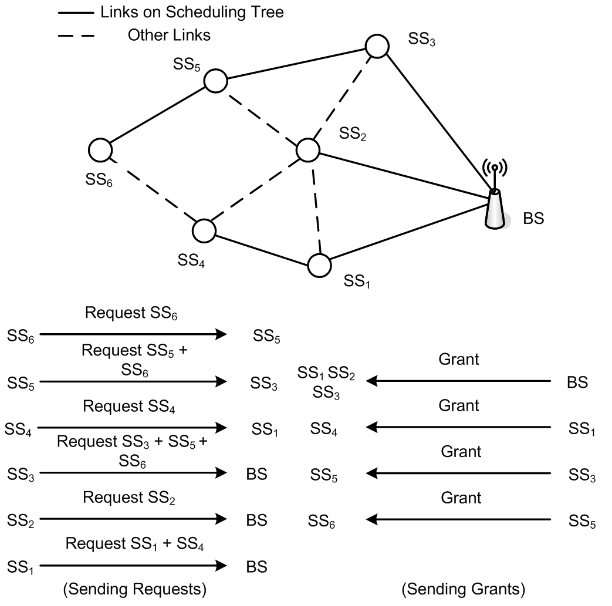

Figure 1: Example of centralized scheduling

delay and packet drops. Our mechanism temporarily uses the intranet traffic part of the network in such scenarios to carry the excess Internet traffic. It strictly abides by the IEEE 802.16 standard and can be implemented without any change in the standard both in BS and SSs.

In the literature, there are various simulation models regarding the PMP mode of WiMAX. However, a detailed system model for the Mesh mode does not exist. In this paper we develop a system model for the Mesh mode of WiMAX using OPNET 11.5 simulation software [2]. We also develop algorithms for the intentionally unstandardized components for our model to operate. These are a BS scheduler and the distributed scheduling requester, which have a significant effect on the overall performance performance of the WiMAX network.

The rest of the paper is organized as follows. In Section II, the scheduling mechanisms of the Mesh operation mode of IEEE 802.16 standard are described. In Section III, proposed CQAR mechanism is explained. This section also entails additional mechanisms developed by the authors for the Mesh operating mode. Simulation environment and results are given in Section IV. Finally, we draw conclusions in Section V.

2. TRAFFIC SCHEDULING IN MESH MODE

Two scheduling methods are defined in the Mesh mode of IEEE 802.16: *Centralized Scheduling* (*CS*) and *Distributed Scheduling* (*DS*). While CS can be used alone, DS can only be used with CS. CS is similar to the PMP mode. Each SS sends its bandwidth request to the BS and all the scheduling in the network is managed by the BS. A node that is not directly connected to the BS send its bandwidth request message to its parent node who forwards it over its own parent node towards BS. Each SS requests bandwidth only for the links on the scheduling tree and only for uplink in these links. This mode is generally used for Internet traffic in the network.

DS entails two methods: *Coordinated Distributed Scheduling* (*CDS*) and *Uncoordinated Distributed Scheduling* (*UDS*). As opposed to CS, none of these methods has a single point of scheduling control. Instead, every SS distributes the scheduling information of its one-hop neighbors and its own scheduling information to its one-hop neighbors. Thus, each node knows the scheduling scheme in its two-hop neighborhood and makes its scheduling accordingly. Both of the methods use a three-way handshake mechanism for bandwidth allocation. The main difference between these two methods is that the scheduling information is sent in a collision free manner in CDS whereas in UDS collisions of scheduling messages are possible. Distributed scheduling is generally used for intranet traffic in the network.

2.1 Centralized Scheduling

In the CS, CSCH messages are used for both requests and grants. Also, the BS informs SSs about the topology of the network using periodic CSCF messages. SSs that do not have any child nodes, send their uplink bandwidth requests to their parent nodes via CSCH messages. SS with child nodes take the uplink bandwidth requests from all their child nodes and generate a CSCH message including these requests as well as their own uplink request. In a network consisting of N SSs, since each CSCH message uses one *Transmission Opportunity* (*TO*), N number of TOs are needed in order for all SSs to send their bandwidth requests to the BS. The control subframe has a maximum size in the standard ($Csto_{max}$), also some of these TOs should be reserved for the CDS.

The BS generates a grant using the uplink bandwidth requests from SSs and the downlink bandwidth requests from itself via a mechanism called the BS scheduler. The grant is then distributed to the SSs again by using CSCH messages. Upon receiving the grant message, SSs that have child nodes forward this grant to their child nodes. This forwarding continues until every node acquires the new grant. Similar to the request sending procedure, the grant messages may exceed one frame.

When sending uplink requests, the SS with the highest Node ID sends its CSCH message first and the SS with the lowest Node ID sends last. Grant messages propagate in the network in the reverse order of the request messages. Starting from the BS, continued by the SS with the lowest Node ID that has at least one child and ends with the SS with the highest Node ID with at least one child (Figure 1). Using the topology information gathered by the CSCF messages together with this request and grant sending sequences, the SSs send their CSCH messages without any collisions.

2.2 Distributed Scheduling

In coordinated distributed scheduling, scheduling information is sent using the DSCH messages in a non-collision manner. Since a single point of control does not exist in CDS, each SS has to calculate a unique TO to send its DSCH message to avoid collisions. All SSs in the network know the complete topology of the network from the periodic CSCF messages. Node ID's of the SSs are also included in this message. In order to choose a unique TO, each SS calculates a hash value for each of its two-hop neighbours including itself using SS's Node IDs and the number of the TO in question (starting from the first applicable TO for itself). For each TO, the SS whose hash value is the greatest is assumed to have won the contention and uses this TO to send its DSCH message. Each SS continues to calculate these hash values until it wins the contention for one TO. In order to be fair among SSs this mechanism introduces an additional backoff mechanism. After winning for a TO, an SS cannot enter contention for a given number of TOs (Xmt Holdoff Time).

Each DSCH message includes requests, availabilities and grants. Every SS sends requests (if necessary) for every distributed link it has in the network when its DSCH message sending TO comes. Upon receiving a DSCH message, a SS checks if there is a request for the link between itself and the transmitting SS. Then, the receiving SS checks whether it can grant the request in several steps. Each SS knows in which minislots of the DS part of the data subframe it is available to transmit and/or receive data. When sending a DSCH message, a SS includes its availabilities into the message. Thus, the granter knows the availabilities of both the requester and itself. Based on these information, the receiving SS generates a grant to the request if a grant can be given to this request. This grant is sent to the requester with a DSCH message. If the requester is still available for this grant according to its current availability list, it sends the grant again to the granter to acknowledge the grant. Thus, the three-way handshake is complete and the data transmission commences as agreed between the two SSs. A recent work by Cao et al. studies the effects of different variables in the performance analysis of distributed scheduling Mesh mode [3].

Uncoordinated distributed scheduling is considerably similar to CDS in many aspects. The same three way handshake and DSCH message format used in CDS is also used for data allocation in UDS. However, the DSCH messages are not sent in the control subframe in UDS. SSs use minislots of distributed scheduling part of the data subframe according to their availability lists to send their DSCH messages. Unlike the CDS, there is no guarantee that the DSCH messages will not collide. If there does not exist any available minislots in the distributed scheduling part of the data subframe for a given SS at a frame, UDS cannot be used. This scheduling method enables rapid and impromptu setup of schedules on a link-by-link basis in the Mesh Mode.

3. CENTRALIZED QUEUE AWARE ROUTING - CQAR

CQAR is a routing-scheduling algorithm which utilizes the potential routing paths for diverging traffic to combat network congestions. In CQAR, if an SS has more than one potential parent node, a second node among the potential parent nodes is selected as the pseudo parent after selecting the parent node. In the case of congestion in the CS, the SS changes the routing of the Internet traffic from its parent node to its pseudo parent node. Since the link between the SS and its pseudo parent node is a distributed link, the centralized link usage for this node's Internet traffic decreases (Figure 2). Thus, the total traffic introduced to the CS decreases, and the system goes back to an uncongested state. In order not to hinder the intranet traffic unnecessarily, the SS switches back its routing of Internet traffic from its pseudo parent to its parent node when the congestion in the CS ends.

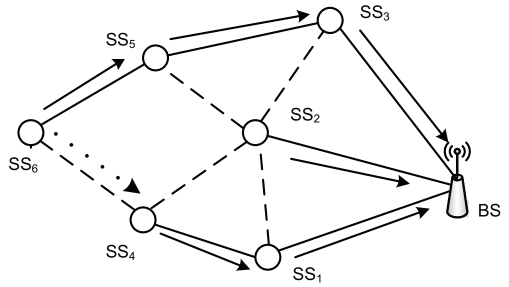

Figure 2: Internet traffic routing (solid arrows: Uncongested, dotted arrow: CS congested SS6 using pseudo parent)

There can be different approaches to determine when to change the route from the parent node to the pseudo parent node, and when to revert back to the parent node from pseudo parent node. In this paper we use a simple yet effective approach that utilizes the the queue length of the centralized link as the indicator for detecting the start of a congestion. When the queue length of this link exceeds a certain threshold (Centralized Link Congestion Threshold, $CLCth$), the SS starts to route all of its Internet traffic from its pseudo parent. The end of the congestion is identified by checking the queue length of the centralized link again. We use a second threshold value (Centralized Link Uncongestion Threshold, $CLUth$) for switching the Internet traffic routing back to the parent node to avoid instability in the system.

Our method only changes the routing of the data packets in the network. We do not propose any changes to the MAC layer mechanisms of IEEE 802.16 standard. Thus, our method can be implemented in current and future SS and BS devices easily since CQAR approach abides by the standards.

3.1 Hop Count Aware BS Scheduler

QoS schedulers in both the BS and SS sides are left unstandardized in the IEEE 802.16 standard. Considering the significant effect that these protocol components may induce on the performance, several BS schedulers have been proposed for the Mesh operating mode in the literature [4, 5, 6]. Since these BS schedulers use Space Division Multiple Access (SDMA) or spatial reuse, several changes should be made to the standard in order for these BS schedulers to work. In addition to the BS schedulers, several routing algorithms have also been developed for the IEEE 802.16 [7, 8, 9, 10].

To handle Internet traffic scheduling, we develop a *Hop Count Aware* (*HCA*) BS Scheduler. Since the CS is mainly used for Internet traffic the requested traffic is destined to the BS. If we denote the hop count of SS_i as Hc_i, the uplink bandwidth request (in bits) of this SS with $ReqUp_i$, and the uplink bandwidth grant (in bits) for this SS with $GrtUp_i$, this grant actually consume, $GrtUp_i \cdot Hc_i$ bits from the network's total bandwidth. Since the BS knows the Hc_i values for each SS in the network, it can calculate how much traffic is actually necessary for each SS ($ReqUp_i \cdot Hc_i$), and use these values instead of the $ReqUp_i$values. The HCA uses this approach in both uplink and downlink grants.

HCA also prioritizes SSs based on their hop counts, namely Hc_i values, starting from SSs with $Hc_i = 1$, and lastly SSs with $Hc_i = Max_{Hc}$ where Max_{Hc} denotes the highest hop count in the network. Thus, the HCA tries to increase the utilization of the network. The total amount of actual bandwidth usage of the SSs with the current hop count is compared with the remaining available minislots in the network. If the total amount is less than or equal to the remaining available minislots, each request is granted completely. Otherwise, the SSs with the current hop count are listed randomly using a uniform distribution, and their requests are checked in the order in this list.

3.2 Distributed Scheduling Requester

While the DSCH packet format and the three-way handshake is defined in the standard, neither a requester nor a granter is defined. We develop a requester that handles the data traffic with low delay and jitter values while reducing the number of requests.

Firstly, our requester records the amount of traffic received from the higher layers during each frame in minislots. When an SS is sending its DSCH message, an SR and Per value is calculated for each distributed link of that SS, based on the recorded received traffic values. In the DSCH message, the SR parameter defines how many minislots a transmission consists of in each frame. The Per parameter defines how many frames this transmission lasts. Then the variance of the last M values of this received traffic are calculated. If the variance is below a certain threshold, the requester sets the Per value equal to M and the SR value to the average of these M values. If the variance is below the threshold, another variance is calculated using the last $\frac{M}{2}$ values and compared with another threshold. This calculation continues unless the variance is below its related threshold, or $\frac{M}{2}$ is equal to 1. After a request is granted, the requester will not try to send any requests for this link until the current request's Per value is expired. Thus, our requester sends many requests with different number of SRs in each request with small Per values for the bursty traffic. On the other hand, if the traffic is stable, a few requests are sent using SR values close to each other on each request with high Per values.

We employ a streamlined granter in the CDS. The granter checks for a vacancy for the request in the two availability lists. It gives a grant if it finds an appropriate vacancy in the next 256 frames.

4. EXPERIMENTAL RESULTS

4.1 Simulation Scenario and Parameters

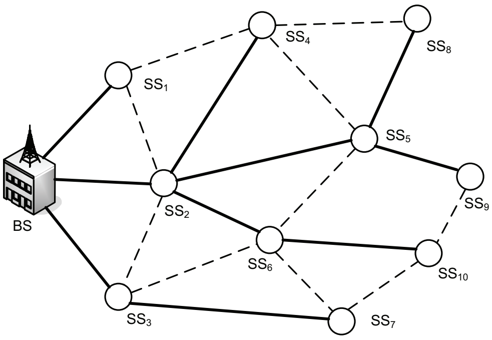

Figure 3: Simulation topology

Table 1: Simulation parameters

System Parameters	
Frame Duration (msec)	5
Minislot Length (OFDM Symbol)	2
Data Subframe Modulation	64-QAM
Data Subframe FEC Rate	3/4
SS Count	10
Queue Size (Centralized Link) (Kbit)	1000
Queue Size (Distributed Link) (Kbit)	250
Max_{Hc}	3
$Csto$ (TO)	15
$Csto_{CS}$(TO)	10
$Csto_{DS}$(TO)	5
$ReqFrCnt$ (Frame)	1
$GrtFrCnt$ (Frame)	1
Minislot Count in Data Subframe	155
Minislot Count Reserved for CS	125
Minislot Count Reserved for DS	30
CS Capacity	44 Mbit/sec
DS Capacity	11 Mbit/sec
$CLCth$	30%
$CLUth$	10%
$DLCth$	40%
Traffic Parameters	
Downlink Traffic Rate per SS (Mbit/sec)	1.0
Uplink Traffic Rate per SS (Mbit/sec)	1.1
Intranet Traffic Rate ($SS_{10} -> SS_9$) (Mbit/sec)	8.8
Intranet Traffic Rate ($SS_7 -> SS_{10}$) (Mbit/sec)	1.0

We test our proposed solution in a moderately dense topology consisting of one BS and 10 SSs (Figure 3). The bold lines denote centralized links whereas the dashed lines denote distributed links. The maximum hop count in the topology is three, and there are three SSs at one-hop distance, four SSs at two-hop distance, and three SSs at three-hop distance.

We assume free space propagation model in the physical layer. Thus, we adopt the least robust PHY layer available in the standard, 64-QAM with 3/4 coding rate, for the data subframe. The frame duration is chosen as 5 msec. To reduce the $ReqFrCnt$ and $GrtFrCnt$ values, we set the $Csto$ to its maximum available value in the standard. The simulation parameters are listed in Table 1.

We assume three traffic types in the simulation: downlink Internet, uplink Internet, and intranet traffic (Table 1). All types of traffic follow Poisson distribution. The downlink Internet traffic is considered as background traffic and runs during the whole simulation. The uplink Internet and the intranet traffics are implemented as on/off traffics with changing starting times and duration among different SSs. Since the distributed scheduling utilizes SDMA, we only introduce intranet traffic to and from the three-hop SSs. The simulation duration is 3 minutes, which results in 36,000 frames.

SSs record the queue lengths of their centralized links once every second in our mechanism. Congestion control checking mechanism runs once in every 5 seconds using the queue lengths recorded in the last 5 seconds.

Since SSs do not enter the network simultaneously in a realistic setting, the periodic checking of centralized link queue length in each node should be asynchronous. In our simulation, SS_i starts using the CQAR system at the $QARch_i^{th}$ second. The $QARch_i$ is calculated for each SS as $QARch_i = i \ mod \ 5$.

The simulation starts with only downlink traffic. After the first second, SSs start generating uplink traffic and after the tenth second all the SSs generate uplink traffic. Thus, the network becomes congested. This congested state lasts until two SSs stop their uplink traffic at 45th second. This congested Internet traffic state occurs twice again in the simulation between 77th- 106th seconds and 150th - 167th seconds. Our method changes the routing of the Internet traffic in the congestion cases. Only the three-hop SSs with pseudo-parent change their route since only they are affected by the congestion.

In order to show the effects of our mechanism in a worst case scenario, we introduce heavily loaded intranet traffic when the Internet traffic is already congested at 25th second.

4.2 Simulation Results

The simulation starts with only downlink traffic. Following the first second, SSs start generating uplink traffic and after the 10th second all the SSs generate uplink traffic. Then the network becomes congested due to this excessive load. This congested state lasts until two SSs stop their uplink traffic at 45th second. This congested Internet traffic state occurs twice again in the simulation between 77th- 106th seconds and 150th - 167th seconds. Our method changes the routing of the Internet traffic in the congestion cases. Only the three-hop SSs with pseudo-parent change their route since only they are affected from congestion.

In order to show the effects of our mechanism in a worst case scenario, we introduce loaded intranet traffic when the Internet traffic is already congested at 25th second. The load of the system on one-hop SSs is minor. The average downlink ETE delay of one-hop SSs does not change between the two methods. For the two-hop SSs, the effect of our mechanism is similar since the load does not affect two-hop SSs as in the case of one-hop SSs. The improvement of our method is still marginal. Quantitatively, CQAR reduces the downlink ETE delays of two-hop SSs by 1-1.5 msecs. Similar to downlink ETE delays, baseline method and CQAR perform analogously regarding the uplink ETE delay values for one-hop and two-hop SSs.

In case of congestion in the centralized links, firstly the three-hop SSs suffer high ETE delays and packet drops. During the three congestion periods the ETE delay of all three three-hop SSs increase dramatically. CQAR decreases the high ETE delay values but its reduction is not stable since after changing the Internet traffic routes SSs 8 and 10 only use the alternative routes until the packets at their centralized link queues are sent and their centralized queue lengths drop below 10 % of their capacity. Then, they switch back their original routes. However, there is still significant improvement in terms of mean ETE delay and its deviation: on average, 12% reduction in mean ETE delay and 19% reduction in ETE standard deviation of three-hop SSs. The effects of CQAR are also apparent in the percentage of packet drops. CQAR decreases the percentage of packet drop for both uplink and downlink Internet traffic at the expense of an increase for the intranet traffic.

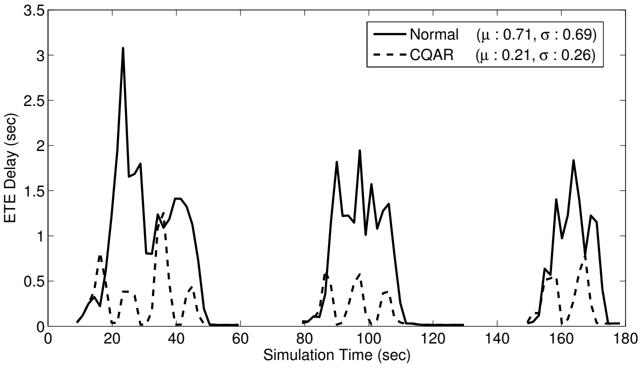

Figure 4: ETE delay of SS_8 (uplink Internet traffic)

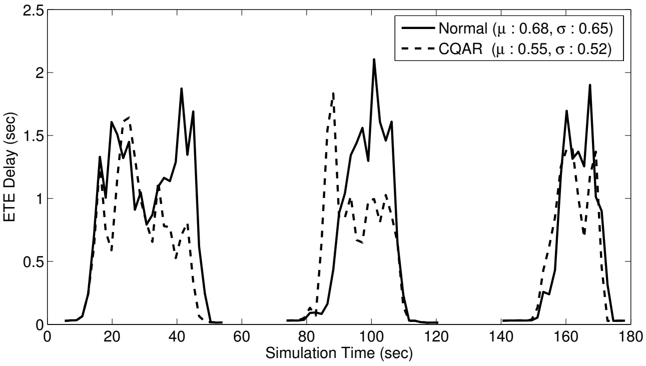

Figure 5: ETE delay of SS_9 (uplink Internet traffic)

Our method works better than the baseline routing method regarding the uplink ETE delay values (Figures 4, 5, and 6). In all figures, μ and σ denote the mean and standard deviation values, respectively. CQAR decreases the delay values of SSs 8 and 10 considerably but because of the reason stated above, it gives instable results. Also, it cannot decrease considerably the uplink ETE delay of SS_9, which has no pseudo-parent.

Unlike Internet traffic, our method increases the ETE delay and number of dropped packets values for the intranet traffic. This side-effect is the main drawback of our method and shown for intranet traffic between SS_{10} - SS_9 in Figure 7. Since CQAR does not check the distributed queue lengths, it cannot know if there is intranet traffic.

5. CONCLUSION & FUTURE WORK

The Mesh operating mode of IEEE 802.16 adds intranet traffic capability and increased area coverage over the PMP operating mode of the standard. In the Mesh mode, the data subframe is divided into two parts for Internet and in-

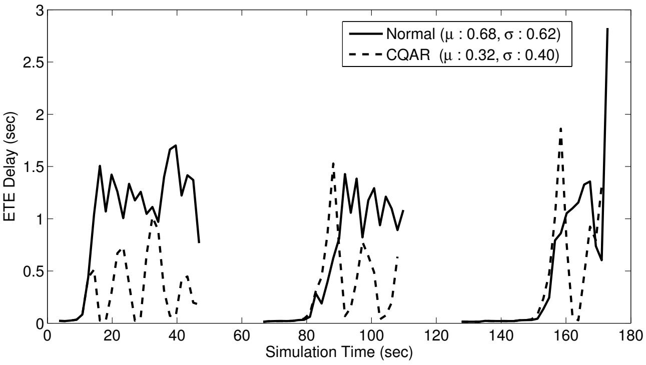

Figure 6: ETE delay of SS_{10} (uplink Internet traffic)

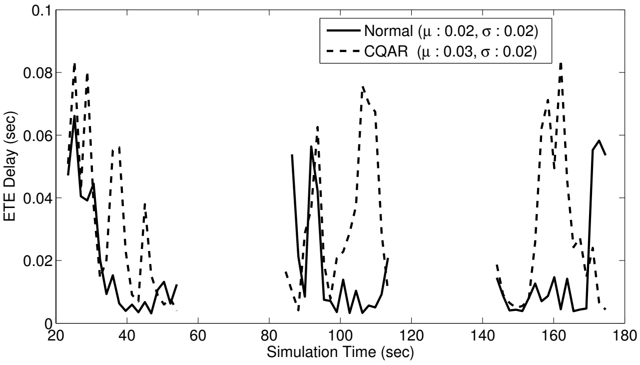

Figure 7: ETE delay (intranet Traffic SS_{10} - SS_9)

tranet traffics, respectively. We develop a cross-layer routing scheduling scheme that exploits links dedicated to intranet traffic for Internet traffic in case of a congestion in the Internet traffic. Since our method does not alter any part of the original IEEE 802.16 standard, it can be implemented to both Mesh BS and SS devices with relative ease.

Our scheme has been tested by extensive simulations. While it improves the systems carrying capacity and decreases the ETE delay of the congested nodes and number of dropped packets, it may go into instable states. Other approaches can also be developed for better system capacity improvement, decision speed, and system stability. A better approach may be to take into account not just the centralized link but also the dynamic states of the exploited distributed links for more stable operation and better performance. For future work we plan to implement this approach and compare its results with the results presented in this paper. Also, we will consider using different traffic types than the poisson traffic used in this paper.

6. ACKNOWLEDGMENTS

This work is partially supported by The State Planning Organization of Turkey (DPT) under Next-generation Satellite Networks and Applications Project, DPT-03 K120250 and Scientific and Technical Research Council of Turkey (TUBITAK) under grant number 104E032.

7. REFERENCES

[1] M. S. Kuran and T. Tugcu. A survey on emerging broadband wireless access technologies. *Elsevier Computer Networks*, 51(11):3013–3046, August 2007.

[2] OPNET. *OPNET Modeler 11.5*. Available at url: http://www.opnet.com.

[3] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu. Modelling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode. In *Proc. ACM 6th Int. Symposium on Mobile Ad Hoc Networking and Computing 2005 (MobiHoc'05)*, pages 78–89, 2005.

[4] D. Kim and A. Ganz. Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems. In *Proc. Int. Conference on Broadband Networks 2005 (BROADNETS'05)*, pages 833–839, 2005.

[5] M. Cao, V. Raghunathan, and P. R. Kumar. A tractable algorithm for fair and efficient uplink scheduling of multi-hop wireless mesh networks. In *Proc. 2nd Int. Workshop on WiMAX Mesh Networks 2006 (WiMesh'06)*, pages 93–100, 2006.

[6] B. Han, W. Jia, and L. Lin. Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks. *Elsevier Computer Communications*, 30(4):782–792, February 2007.

[7] J. Tao, F. Liu, Z. Zeng, and Z. Lin. Throughput enhancement in WiMAX mesh networks using concurrent transmissions. In *Proc. Int. Conference on Wireless Communications, Networking and Mobile Computing 2005 (WCNM'05)*, pages 871–874.

[8] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas. Interference-aware IEEE 802.16 WiMAX mesh networks. In *Proc. 61st IEEE Vehicular Technology Conference 2005 (VTC'05)*, pages 3102–3106, 2005.

[9] H. Shetiya and V. Sharma. Algorithms for routing and centralized scheduling in IEEE 802.16 mesh networks. In *Proc. IEEE Wireless Communications and Networking Conference 2006 (WCNC'06)*, pages 147–152, 2006.

[10] T.-C. Tsai and C.-Y. Wang. Routing and admission control in IEEE 802.16 distributed mesh networks. In *Int. Conference on Wireless and Optical Communications Networks 2007 (WOCN'07)*, pages 1–5, July 2007.