Adaptive Modulation-based TCP-Aware Uplink Scheduling in IEEE 802.16 Networks

Hemant Kumar Rath and Abhay Karandikar
Information Networks Lab, Department of Electrical Engineering
Indian Institute of Technology Bombay, Mumbai 400 076, India
Email: {hemantr, karandi}@ee.iitb.ac.in

Vishal Sharma
Metanoia Inc., 888 Villa Street, Suite 500
Mountain View, CA, 94041-1259, USA
Email: v.sharma@ieee.org

***Abstract*— In this paper we propose polling based uplink scheduling schemes for TCP based applications in a multipoint-to-point *fixed* broadband IEEE 802.16 network. Our schemes adapt the transmission rates between Subscriber Stations (SSs) and the Base Station (BS) dynamically using adaptive modulation. We ensure fairness among the SSs via a credit-based approach in which an SS that misses a chance to transmit due to bad channel gets more weightage when the channel favors scheduling. We also propose a method to compute an optimal polling interval that aims to maximize slot utilization and TCP throughput. We demonstrate through exhaustive simulations that the proposed schedulers maximize link utilization, provide long-term fairness and minimize contraction of TCP congestion window. Implementation of the proposed schemes requires a cross-layer based feedback protocol stack at the BS and SSs.**

I. Introduction

IEEE 802.16-2004 [1], [2], sponsored by the IEEE LAN/MAN society, is a fixed Broadband Wireless Access (BWA) standard for both multipoint-to-point and mesh mode of operation[1]. It defines the Physical (PHY) and Medium Access Control (MAC) layers of the protocol stack. The Physical layer supports both fixed as well as adaptive modulation techniques in the uplink and in the downlink directions. Maximum attainable data rates depend upon the modulation schemes used and the condition of the channel. The MAC layer of IEEE 802.16 is connection-oriented in which each traffic flow between a Subscriber Station (SS) and the Base Station (BS) can be identified by an unique Connection ID (CID). Each flow may fall into one of the four different kinds of services; Unsolicited Grant Service (UGS), Real Time Polling Service ($rtPS$), Non Real Time Polling Service ($nrtPS$) and Best Effort (BE) service. Guaranteed bandwidth in terms of a minimum reserved traffic rate is the basic Quality of Service (QoS) parameter defined at the MAC layer for UGS, $rtPS$ and $nrtPS$ services, whereas it is not so for BE service.

Currently, many Internet applications that belong to BE services of IEEE 802.16 are based on Transmission Control Protocol (TCP). Since TCP is a greedy protocol, there is a need for fair resource allocation scheme to assign resources among the contending TCP flows. When the maximum data rates between SSs and the BS are different, assignment of resources among contending flows becomes critical. We, therefore, propose adaptive modulation-based uplink scheduling schemes for applications based on TCP in a multipoint-to-point IEEE 802.16 network. The first scheme uses only the congestion window ($cwnd$) of the contending flows, whereas the second scheme uses both $cwnd$ and TCP timeouts of the contending flows to allocate resources. The proposed uplink scheduling schemes operate at the BS, which assign time slots and attempt to maximize the use of allocated time slots taking the random nature of the wireless channel into consideration. We introduce a credit-based approach using deficit counters to ensure long-term fairness among the SSs.

[1]In this paper, we do not consider mobility.

A. Related Work

IEEE 802.16 network elements are permitted to implement their own scheduling algorithms at the BS for both uplink and downlink as the standard does not specify any specific algorithm to be implemented. Since the BS has knowledge of all queues assigned to SSs and arrival times of packets in the downlink, scheduling is simpler in the downlink. In downlink scheduling, the BS can use a scheduler similar to that used in traditional wired networks like Weighted Fair Queuing (WFQ) [3], Self-Clocked Fair Queueing (SCFQ) [4], Worst-case Fair Weighted Fair Queuing (WF^2Q) [5]. In uplink scheduling, schemes like WFQ, SCFQ and WF^2Q would require computation of virtual start time and finish time at the BS for each packet arriving at SS. Since the packet arrival information is not available at the BS, such schemes are not suitable for uplink scheduling, instead variants of Round Robin Scheduler are the candidates for uplink scheduling.

Most existing schedulers for IEEE 802.16 networks have been designed for $rtPS$ and $nrtPS$ services rather than for BE services. In [6], [7], the authors have analyzed the QoS support at the MAC layer by providing differentiated services to applications with different QoS requirements such as VoIP and web services. They have used Weighted Round Robin (WRR) for uplink and Deficit Round Robin (DRR) for downlink scheduling. In [8], the authors propose an adaptive queue aware uplink bandwidth allocation scheme for $rtPS$ and $nrtPS$ services. The bandwidth allocation is adjusted dynamically according to the variations in traffic load and/or the channel quality. In [9], we have proposed a credit-based scheduling scheme which polls SSs in an optimal manner to address the delay requirements of various classes of service.

B. Motivation and Primary Contribution

The primary contribution of this paper is to propose a fair adaptive modulation-based uplink scheduling scheme for applications based on TCP in IEEE 802.16. Since, the TCP congestion window size ($cwnd$) changes only after one RTT, $cwnd$ is an indication of the number of time slots required per Round Trip Time (RTT). Hence, instead of assigning equal number of slots to all users, we argue that the BS should assign slots in proportion to their $cwnd$, i.e., as per the flow's requirement. Assigning time slots based only on $cwnd$ will result in unfairness among the TCP flows, since flows with smaller RTTs will have larger window size as compared to the flows with larger RTT. To avoid this unfairness, we introduce a credit-based approach that ensures fairness among the flows. More slots are assigned to the flows which are closer to their TCP timeout, thereby preventing their congestion window from dropping to one due to timeout. By introducing adaptive modulation, fairness measure that only considers slots assigned becomes irrelevant, rather, fairness in terms of amount of data transmitted in a frame should be considered. Hence, we measure fairness on the amount of data transmitted by SSs.

The rest of the paper is organized as follows. In Section II, we discuss our system model. In Section III, we propose two uplink scheduling schemes for TCP based applications. We describe a cross-layer based feedback protocol to implement these schemes, a method to compute the polling interval and long-term fairness in Section IV. In Section V, we describe the experiments and discuss our simulation results. Finally, we provide the concluding remarks and scope of the future work in Section VI.

II. System Model

We consider a multipoint-to-point IEEE 802.16 based network where multiple SSs are connected to a centralized BS as shown in Fig. 1. We consider WirelessMAN-SC air interface as an example, which supports both fixed and adaptive modulation in uplink and downlink directions. Based on the channel condition, the BS selects a modulation scheme to be used and informs to the SS, such that data can be transfered reliably between the BS and SS. In the downlink, QPSK and 16-QAM are the mandatory and 64-QAM is the optional modulation scheme, whereas in the uplink QPSK is the mandatory modulation scheme and 16-QAM and 64-QAM are optional modulation schemes. Since QPSK is mandatory in the uplink, we use QPSK for fixed modulation in this paper. The standard also allows three broad channel bandwidths (B) namely, 20 MHz, 25 MHz and 28 MHz.

Though the standard defines maximum baud rate, modulation schemes to be used and maximum data rate possible for WirelessMAN-SC category, it does not specify the Signal to Noise Ratio (SNR) thresholds for choosing different modulation schemes to be used. The maximum data rate attainable for an Additive White Gaussian Noise (AWGN) channel can be expressed as: $R = B \times \log_2(1 + MI \times SNR)$, where R is the maximum attainable data rate and MI is the modulation index. Note that $MI = \frac{-\phi_1}{\log(\phi_2 BER)}$, where ϕ_1 and ϕ_2 are

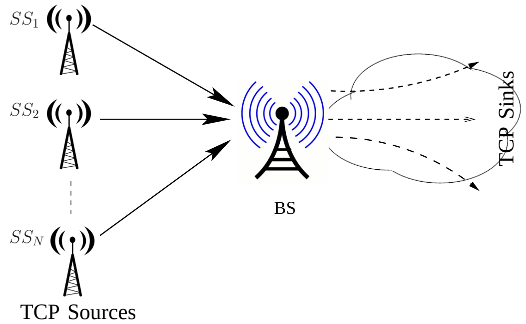

Fig. 1. Multipoint-to-Point Framework in IEEE 802.16 Network

TABLE I

MODULATION SCHEMES IN THE UPLINK OF WIRELESSMAN-SC IEEE 802.16 (CHANNEL BANDWIDTH $B = 25MHz$)

Modulation Scheme	Data Rate R (Mbps)	$\frac{R}{B}$ (bps/Hz)	SNR_{th} (dB) $BER = 10^{-5}$	SNR_{th} (dB) $BER = 10^{-6}$
QPSK	40	1.6	11.27	12.18
16-QAM	80	3.2	17.33	18.23
64-QAM	120	4.8	23.39	24.14

constants depending upon the modulation schemes used [10]. Since the standard specifies fixed data rates to be used, for a particular modulation scheme, SNR thresholds should satisfy:

$$\begin{aligned} SNR &= \frac{2^{\frac{R}{B}} - 1}{MI} \\ &= \frac{(1 - 2^{\frac{R}{B}}) \times \ln(5p_b)}{1.5}, \text{ if } \frac{R}{B} < 4 \\ &= \frac{(1 - 2^{\frac{R}{B}}) \times \ln(0.5p_b)}{1.5}, \text{ if } \frac{R}{B} \geq 4, \end{aligned} \tag{1}$$

where p_b is the target bit error rate. Using (1), the computed SNR threshold for target BERs of 10^{-5} and 10^{-6} for a channel bandwidth of 25 MHz are given in Table I. The normalized data rate $\frac{R}{B}$ for QPSK modulation scheme is 1.6, whereas it is 3.2 and 4.8 for 16-QAM and 64-QAM respectively.

Though the MAC layer of IEEE 802.16 supports both Time-Division Duplex (TDD) and Frequency-Division Duplex (FDD), we consider only TDD. In TDD, time is divided into frames, each of which in turn consists of an uplink subframe and a downlink subframe. Each subframe is composed of a fixed number of slots. The standard supports a bandwidth request-grant mechanism in which bandwidth requests are conveyed either in a contention mode or in a contention-free polling mode. We consider a contention-free polling mode in which the BS polls each SS for its bandwidth requirement.

In our framework SSs are the TCP sources who transmit to the end users (TCP sinks) through the BS. We consider a single TCP flow between each SS and the BS. A set I of TCP flows (also known as source-sink pairs) shares a network of L unidirectional links through the BS. We assume that the links between the SSs and the BS are the bottleneck links of the network whereas the downlink does not have any bandwidth constraint. The capacity of the individual link l is c_l, $l \in L$. Link capacity c_l is a function of the channel condition of the link l. For successful reception, SNR at the receiver should be greater than the minimum SNR threshold (SNR$_{th}$) required among all modulation schemes. Note that from Table I the minimum SNR value for a BER of 10^{-5} is 11.27 dB and for a BER of 10^{-6} is 12.18 dB, which requires QPSK modulation to be used.

III. Uplink Scheduling Schemes

A. TCP Window-Aware Uplink Scheduler with Adaptive Modulation (TWUS-A)

The TCP Window-Aware Uplink Scheduler is a polling based system wherein the BS polls each SS to determine its resource requirement in terms of number of slots required to transmit. Polling can be done once in every frame or in multiple frames. In the proposed scheme, the BS polls each SS periodically, once every k frames. The determination of the value of k is explained in Section IV. An SS with non-zero congestion window size and having SNR greater than the SNR$_{th}$ (corresponding to QPSK modulation) conveys its slot requirement to the BS. The list of SSs that responds to the polling with $cwnd$ size constitutes a *schedulable set* (L_{sch}) at the BS. The BS does not alter set L_{sch} till the next polling opportunity, k frames latter. In subsequent frames (scheduling instants), the BS checks SNR of every user only among the set L_{sch} and schedules those users whose SNR is above SNR$_{th}$. The set of users which can be scheduled during a frame is called an *active set* (L_{active}), which is a subset of the set L_{sch}. The relationship between polling interval and scheduling instances is shown in Fig. 2. In every frame the BS schedules the SSs belonging to the set L_{active} based on a variant of Deficit Round Robin [11] scheduler described in the following paragraph. In this scheme the BS computes the weight $W_i(n)$ of each active SS_i in each frame n and then assigns slots in proportion to its weight. The weight of each SS is updated on a frame by frame basis and is computed in the following manner.

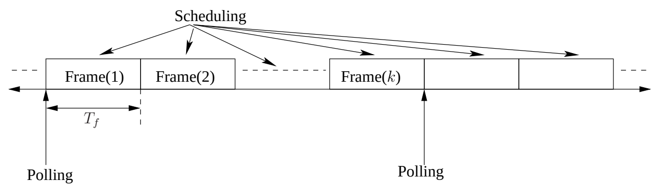

Fig. 2. Uplink Scheduling in IEEE 802.16

Let M be the number of subscriber stations in set L_{sch}. Let $cwnd_i$ be the congestion window size of SS_i which is conveyed to the BS at the time of polling. Let N_s be the total number of uplink data slots in a frame of length T_f. We assume that the number of uplink slots available in a frame is much larger than the number of schedulable subscriber stations. Let $R_i(n)$ and $N_i(n)$ be the rate of transmission and number of physical slots used by SS_i in frame n respectively. At the start of the system (system initialization), we compute quantum size which is an indication of the average amount of data transmission by each schedulable user in a frame as: $Q(0) = \frac{R_{min} N_s T_s}{M}$, where R_{min} is the minimum rate of transmission (corresponds to QPSK in our case) and T_s is the length of each time slot. In each subsequent frame n we update the quantum size as:

$$Q(n) = \frac{1}{M} \sum_{i \in L_{sch}} R_i(n-1) \times N_i(n-1) \times T_s \qquad (2)$$

To keep track of the amount of data transmitted by SS_i as compared to the quantum size $Q(n)$ and to provide fairness among the subscriber stations, the BS maintains a deficit counter for each SS. At the beginning of a flow (or at the connection setup), the deficit counter DC_i of SS_i is initialized to zero. The deficit counter $DC_j(n)$ of each $SS_j \in (L_{sch} \setminus L_{active})$ is *incremented* by $Q(n)$, the amount of service it has missed since it is not scheduled due to bad channel. Likewise, the deficit counter $DC_i(n)$ of $SS_i \in L_{active}$, that has received more than its minimum share $Q(n)$ of the uplink slots is *decremented* by the amount of service that SS_i received over and above its quantum $Q(n)$. The deficit counter of SS_i is updated at the scheduling instant n as:

$$\begin{aligned} DC_i(n) &= DC_i(n-1) \\ &\quad + \frac{\sum_{j \in L_{sch}} R_j(n-1) \times N_j(n-1) \times T_s}{M} \\ &\quad - R_i(n-1) \times N_i(n-1) \times T_s \end{aligned} \qquad (3)$$

From (3), we observe that depending on the number of slots assigned in the present frame as well as in the previous frames, the deficit counter can become positive or negative. Hence, we appropriately scale the deficit counter to obtain $dc_i(n)$ by adding the magnitude of the minimum deficit counter value among set L_{active} to the deficit counter $DC_i(n)$. In other words,

$$dc_i(n) = DC_i(n) + \min_j |DC_j(n)|, \forall j \in L_{active}. \qquad (4)$$

At the start of a flow (or at the connection setup), the scaled deficit counter dc_i is initialized to one. Though $cwnd_i$ for SS_i is fixed for an RTT (which is captured by the polling at the start of each polling interval), the demand (requirement) $D_i(n)$ varies from frame to frame as a result of scheduling and transmission, and can be expressed as follows:

$$\begin{aligned} D_i(n) &= cwnd_i \times PL - Tx_i(n-1) \\ &= D_i(n-1) - N_i(n-1) \times R_i(n-1) \times T_s, \end{aligned} \qquad (5)$$

where PL is packet length in bits (packets are of fixed length) and $Tx_i(n-1)$ is the total number of bits transmitted by SS_i from the polling instant to the current scheduling instant. $Tx_i(n-1) = 0$, at the start of the polling interval $\forall i \in I$. The scaled deficit counter and the weights are computed only for users belonging to set L_{active}. For all other users, the weights are zero. The BS determines the weight $W_i(n)$, $\forall i \in L_{active}$ in frame n using the following equation:

$$W_i(n) = \frac{\frac{D_i(n)}{R_i(n)} \times \frac{dc_i(n)}{R_i(n)}}{\sum_{j \in L_{active}} \frac{D_j(n)}{R_j(n)} \times \frac{dc_j(n)}{R_j(n)}}. \qquad (6)$$

Equation (6) essentially computes a weight $W_i(n)$ in frame n that is directly proportional to the normalized (by $R_i(n)$)

product of the scaled deficit counter and demand. In traditional TCP, if a flow has small RTT, its $cwnd$ is large. Allocating time slots in proportion to $D_i(n)$ (or $cwnd_i$) may result in assigning even larger number of time slots to such flows. The credit-based approach here ensures that the scaled deficit counter value and hence weights for such flows will be small and thereby ensures fairness. After the computation of weights, the BS assigns slots to $SS_i, \forall i \in L_{active}$ in frame n using:

$$N_i(n) \quad = \quad \frac{1}{T_s} \times \min\left(\frac{W_i(n) \times T_f}{\sum_{j \in L_{active}} W_j(n)}, \frac{D_i(n)}{R_i(n)}\right). \quad (7)$$

The first term in the bracket of (7) corresponds to the number of slots as per the weight $W_i(n)$ while the second term corresponds to the number of slots as per the demand $D_i(n)$ of SS_i. As discussed before, if a TCP source does not get an acknowledgment before the TCP timeout occurs, it drops its congestion window to one. TCP timeout occurs usually due to congestion in a link, but can also occur due to a TCP unaware scheduling process. For example, the number of slots assigned to an SS may not be enough to transmit the window of data in one RTT resulting in TCP timeout. To avoid this scenario, we propose a Deadline based TCP Window-Aware Uplink Scheduler (DTWUS-A) in the next section.

B. Deadline based TCP Window-Aware Uplink Scheduler with Adaptive Modulation (DTWUS-A)

In this scheme, we use TCP timeout information along with the $cwnd$ and the deficit counter value to compute the weights. An active SS whose TCP flow is approaching TCP timeout is scheduled with a larger weight than others[2]. We define deadline d_i for SS_i as the amount of time that it can wait before reaching TCP timeout since its last scheduling instant. At the start of a connection, d_i of SS_i is initialized to TTO_i (TCP timeout of SS_i). If SS_i is scheduled in a frame n, then the deadline $d_i(n)$ remains same as $d_i(n-1)$. Else, $d_i(n)$ is decremented by one frame duration from its previous value. In other words, at the n^{th} frame deadline is updated as:

$$d_i(n) = d_i(n-1) - T_f, \quad (8)$$

If T_f exceeds $d_i(n-1)$, then the deadline $d_i(n)$ is initialized to TTO_i. In that case, the TCP flow experiences a timeout before it gets scheduled, resulting in its congestion window dropping to one. SS_i will start retransmitting again with a $cwnd$ of one and a fresh timeout value. The deadline introduced here is a measure of how close a TCP flow is to its TCP timeout. After computing the scaled deficit counters as in (4) and deadlines as in (8), the BS determines the weight $W_i(n)$ of SS_i, $\forall i \in L_{active}$ in frame n using the following equation:

$$W_i(n) = \frac{\frac{D_i(n)}{R_i(n)} \times \frac{dc_i(n)}{R_i(n)} / d_i(n)}{\sum_{j \in L_{active}} \frac{D_j(n)}{R_j(n)} \times \frac{dc_j(n)}{R_j(n)} / d_j(n)}. \quad (9)$$

[2]TCP flows generally start at random and hence different flows have different residual times to reach TCP timeout.

Equation (9) is similar to (6) except for the new term deadline $d_i(n)$. The use of the deadline in the weight computation ensures that the weight of a user that has a smaller deadline is higher as compared to that of a user that has a larger deadline. After the computation of weights, the number of slots assigned to SS_i, $\forall i \in L_{active}$ in frame n is computed using (7). The pseudo-code of the proposed schedulers TWUS-A and DTWUS-A is presented in Algorithm 1. We have combined both schedulers by using $Flag_{deadline}$, which is set to one for DTWUS-A and is set to zero for TWUS-A.

IV. Implementation and Fairness Measure

The block diagram of the proposed uplink scheduler is shown in Fig. 3. Each SS while sending the bandwidth request sends the current congestion window and TCP timeout value to the BS. The BS in turn, computes the number of slots to be assigned and decides the modulation scheme to be used by each SS and conveys this information to each SS through the uplink map. The scheduling is done at the MAC layer of the BS with the help of PHY layer information like SNR between the BS and SSs and TCP layer information like $cwnd$ and TTO at SSs.

We argue that the polling interval k should be the minimum RTT[3] among all TCP flows going through the BS. This is because, the TCP timeout value is typically chosen to be four to five times the RTT in most TCP implementations. Therefore, if we choose the polling interval to be equal to two RTTs, then any SS with an ongoing TCP flow that misses polling needs to be polled at the next opportunity (as the TCP flow of that SS might be reaching TCP timeout). Similarly, if the polling interval is more than two RTTs, and if the BS misses one SS with an active TCP flow, then congestion window reduction for that TCP flow will likely occur with high probability. This is because, the chance of not getting scheduled in the next opportunity before TCP timeout is very high. If polling is very frequent, i.e., more than once per RTT, then more control slots will be spent for polling. Moreover one does not gain due to frequent polling, since the congestion window itself changes after one RTT. Hence, we choose a polling interval to be equal to the minimum RTT of the active TCP flows.

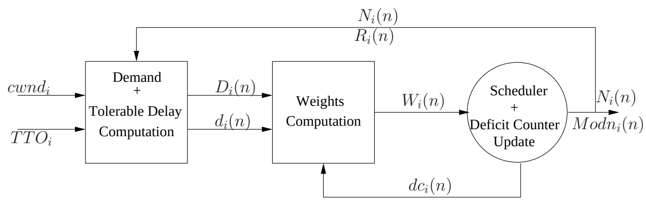

Fig. 3. Block Diagram of the Proposed Uplink Scheduler

A. Discussions on Fairness Measure

In the proposed schemes, we assign more slots to an SS having a bad channel than an SS with a better channel

[3]Typical TCP RTTs are in the range of 100 msec - 200 msec, whereas the frame length T_f in IEEE 802.16 is either 0.5 msec or 1 msec or 2 msec.

by using adaptive modulation techniques. By using deficit counters in weight computations, we ensure that, each SS gets equal opportunity in terms of the amount of data transmitted over time. The proposed scheduler enforces fairness by not allowing greedy users to increase their congestion windows. In addition to the deficit counters, by choosing the polling interval to be equal to one RTT, we provide more opportunity for the SSs to be polled by the BS. Let $Tx_i(t)$ be the amount of data transmitted by SS_i in time interval $[0, t]$, $i \in I$, the set of users. For the scheduling scheme to be long-term fair, it can be argued that the following equation holds:

$$\lim_{t\to\infty} \frac{Tx_1(t)}{t} = \cdots = \lim_{t\to\infty} \frac{Tx_i(t)}{t} = \cdots = \lim_{t\to\infty} \frac{Tx_n(t)}{t} \quad (10)$$

The proof of (10) is omitted here due to space constraint.

V. Experimental Setup and Performance Evaluation

We have simulated an IEEE 802.16 multipoint-to-point network as shown in Fig. 1 with one BS and 10 SSs. We simulate one TCP flow per SS. The TCP flows are started randomly and the RTTs of the flows are updated using exponential averaging. The random channel gains between SSs and the BS are log-normally distributed with variance σ=8 dB. Each SS has a single buffer of infinite size. The frame duration T_f is set equal to 2 msec[4]. The uplink subframe T_{ul} consists of 500 data slots (assuming negligible control slots). We consider both equal and unequal distances between SSs and the BS. For equal distances, the distances of all SSs from the BS are 1 km each and for unequal distances the distances between SSs (SS_1 - SS_{10}) and the BS are 0.90 km, 1.00 km, 1.10 km, 0.90 km, 0.95 km, 1.10 km, 1.00 km, 1.00 km, 1.10 km and 1.01 km respectively. We have conducted four sets of experiments based on distances and proposed schedulers TWUS-A and DTWUS-A. We have also conducted another four sets of experiments based on distances and deadline using fixed modulation scheme QPSK. The algorithms are named as TWUS and DTWUS (corresponds to TWUS-A and DTWUS-A with fixed modulation) in this case. We have used a discrete event simulator. The system parameters used in this paper are presented in Table II.

TABLE II
System Parameters

Type	Parameters
Channel Bandwidth	25 MHz
Adaptive Modulation Schemes	QPSK, 16-QAM, 64-QAM
Bit Error Rate	10^{-6}
Path Loss Factor (γ)	4
Number of Frames Simulated	25000
TCP Type	TCP Reno
Number of Independent Runs	10

A. Results

The number of slots allocated to various SSs placed at equal as well as unequal distances from the BS using adaptive modulation is shown in Table III. We observe that the number of slots assigned with equal distances is more uniform as compared to unequal distances case. We also observe that the slot assignment using DTWUS-A scheduler is fair as compared to that of TWUS-A scheduler. Tables IV and V show the $cwnd$ variation among the SSs for various cases. We observe that the average window size achieved by adaptive modulation is larger by 32% - 36% as compared to the fixed modulation. We also observe that the deadline based scheduler (DTWUS or DTWUS-A) achieves larger window size than the non deadline based scheduler (TWUS or TWUS-A) as the deadline based scheduler attempts to avoid TCP timeouts resulting in larger average congestion window.

From Tables VI and VII, we observe that the average rate of transmission with adaptive modulation scheme is around 75% higher than that of fixed modulation scheme. We also observe that the average transmission rate achieved by an SS depends upon the distance from the BS. So, to achieve fairness in the amount of data transmitted, SSs with lower transmission rate should get more slots compared to SSs with higher transmission rate. This is illustrated in Table VIII. Also the total amount of data transmission of deadline based scheduler (DTWUS or DTWUS-A) is more than that transmitted by schedulers without deadline (TWUS or TWUS-A). When the SSs are at unequal distances from the BS, the total amount of data transmitted is less as compared to when the SSs are equidistant from the BS. This is because the proposed schemes are primarily designed for long-term fairness rather than for achieving high sum-capacity.

B. Fairness and Usage of Resources

To assess the fairness of our proposed scheduling schemes, we compute the Jain's Fairness Index (JFI) [12] for the amount of data transmitted by each SS. This is illustrated in Table IX. We observe that the JFI is more than 99% when the distance between the SSs and the BS are equal and more than 98% when the distances are unequal. This illustrates that our scheduling schemes are fair. We also analyze the slot usage of our proposed schedulers as shown in Table IX. We observe that the usage of slots is more than 96% in most of the cases. Moreover, the schedulers based on fixed modulation scheme (QPSK) have more slot usage than schedulers based on adaptive modulation scheme. Even though adaptive modulation results in increasing transmission rate by around 75%, average $cwnd$ is increased only by 36% resulting in smaller slot usage. Usage of slots can be further increased by adding different classes of traffic along with TCP traffic.

To analyze the fairness of our proposed scheduling schemes with different log-normal fading, we have simulated the schemes with five different σ (2, 4, 6, 8 and 10 dB). The results are plotted in Fig. 4 and Fig. 5. We observe from these figures that the proposed scheduling schemes are fair even for a large variation of fading in the channel.

VI. Concluding Remarks

In this paper, we have proposed adaptive modulation-based fair uplink scheduling schemes for applications based on TCP in a multipoint-to-point IEEE 802.16 network. We have

[4]Frame duration (T_f) is equally divided between uplink subframe (T_{ul}) and downlink subframe (T_{dl}).

considered TCP congestion windows, TCP timeout values and the channel condition between the BS and SSs for scheduling. We have attempted to avoid TCP timeouts occurring due to TCP un-aware scheduling at the MAC layer. The proposed schemes succeed in stabilizing the congestion window variation. With adaptive modulation, we have achieved higher rate of transmission as compared to fixed modulation (QPSK). We have demonstrated through exhaustive simulations that fairness in slot assignment and in amount of data transmitted is achievable. Though, for simulation purposes, we have considered fixed broadband WirelessMAN-SC as an example, the framework reported in the paper can easily be extended to OFDM and OFDMA based mobile broadband. We are currently investigating in this direction.

REFERENCES

[1] LAN/MAN Standards Committee, *IEEE Standards for Local and Metropolitan Area Network: Part 16: Air Interface for Fixed Broadband Wireless Access Systems*. IEEE Computer Society and IEEE Microwave Theory and Techniques Society, May 2004.

[2] "Understanding Wi-Fi and WiMAX as Metro-Access Solutions," *Intel Corporation, White Paper*, August 2004. Available at: http://whitepapers.silicon.com. Accessed on:10/10/2005.

[3] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," in *ACM SIGCOMM Symposium on Communications Architectures and Protocols*, September 1999.

[4] S. Golestani, "A Self-Clocked Fair Queuing Scheme for Broadband Applications," in *Proc. IEEE INFOCOM*, pp. 636–646, 1994.

[5] J. C. R. Bennett and H. Zhang, "WF^2Q: Worst-case Fair Weighted Fair Queueing," in *Proc. IEEE INFOCOM*, pp. 120–128, 1996.

[6] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi, "Performance Evaluation of the IEEE 802.16 MAC for QoS Support," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 26–38, January 2007.

[7] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of Service Support in IEEE 802.16 Networks," *IEEE Network*, vol. 20, pp. 50–55, March-April 2006.

[8] D. Niyato and E. Hossain, "Queue-Aware Uplink Bandwidth Allocation and Rate Control for Polling Service in IEEE 802.16 Broadband Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 668–679, June 2006.

[9] H. K. Rath, A. Bhorkar, and 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 *Proc. of IEEE Globecom*, November 2006.

[10] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[11] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 375–385, June 1996.

[12] R. Jain, D.-M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *Technical Report TR-301, DEC Research Report*, September 1984.

TABLE III

AVG. NO. OF SLOTS ASSIGNED (X10^5) (ADAPTIVE MODULATION)

SS	Equal Distances		Unequal Distances	
No.	TWUS-A	DTWUS-A	TWUS-A	DTWUS-A
1	12.03	12.14	11.53	11.54
2	11.99	12.08	12.08	12.24
3	12.03	12.17	12.26	12.48
4	12.05	12.09	11.52	11.44
5	12.03	12.13	11.78	11.96
6	11.97	12.14	12.40	12.36
7	12.02	12.13	12.06	12.20
8	12.02	12.06	12.07	12.23
9	12.00	12.13	12.26	12.47
10	12.05	12.10	12.13	12.15

TABLE IV

AVERAGE WINDOW SIZE (ADAPTIVE MODULATION)

SS	Equal Distances		Unequal Distances	
No.	TWUS-A	DTWUS-A	TWUS-A	DTWUS-A
1	20.82	22.44	21.66	22.89
2	21.46	22.12	21.28	22.21
3	21.01	22.56	20.27	22.11
4	21.25	22.04	21.86	22.93
5	21.43	22.17	21.67	22.85
6	21.16	22.38	20.67	21.34
7	21.11	22.85	21.31	22.70
8	21.50	22.14	21.18	22.35
9	21.24	22.18	20.67	21.99
10	21.31	22.55	21.75	22.51

TABLE V

AVERAGE WINDOW SIZE (FIXED MODULATION)

SS	Equal Distances		Unequal Distances	
No.	TWUS	DTWUS	TWUS	DTWUS
1	16.15	16.74	16.21	17.02
2	16.06	15.85	15.96	16.41
3	16.85	16.07	16.10	15.09
4	16.40	17.02	17.62	17.58
5	16.26	16.56	16.24	16.66
6	15.82	16.54	14.96	16.14
7	15.94	16.00	16.14	16.44
8	15.75	16.64	16.33	16.43
9	15.84	16.12	15.29	16.21
10	16.24	16.42	15.72	16.25

TABLE VI

AVG. TRANSMISSION RATE ($Mbps$) (ADAPTIVE MODULATION)

SS	Equal Distances		Unequal Distances	
No.	TWUS-A	DTWUS-A	TWUS-A	DTWUS-A
1	56.45	56.41	66.94	66.88
2	56.43	56.48	56.41	56.58
3	56.45	56.69	46.91	47.02
4	56.33	56.51	66.96	66.91
5	56.43	56.40	61.59	61.52
6	56.62	56.42	46.89	47.10
7	56.41	56.38	56.43	56.39
8	56.49	56.53	56.43	56.49
9	56.48	56.41	47.04	47.21
10	56.49	56.52	55.52	55.52

TABLE VII

AVG. TRANSMISSION RATE ($Mbps$) (FIXED MODULATION)

SS	Equal Distances		Unequal Distances	
No.	TWUS	DTWUS	TWUS	DTWUS
1	32.17	32.13	34.98	34.97
2	32.12	32.13	32.09	32.13
3	32.14	32.11	28.93	28.86
4	32.09	32.13	34.94	35.00
5	32.09	32.15	33.59	33.62
6	32.08	32.10	28.91	28.87
7	32.18	32.13	32.13	32.13
8	32.18	32.14	32.11	32.14
9	32.15	32.20	28.92	28.95
10	32.14	32.16	31.87	31.85

Algorithm 1 :TCP Window-Aware Uplink Scheduler with Adaptive Modulation for IEEE 802.16

$DC_i(0) \leftarrow 0\ \forall i$
$dc_i(0) \leftarrow 1\ \forall i$
$N_i(0) \leftarrow 0\ \forall i$
Frame number $n \leftarrow 1$
while TRUE **do**
  Determine L_{sch} for the current polling interval
  Update TTO_i
  if $n = 1$ **then**
    $d_i(0) \leftarrow TTO_i\ \forall i$
  end if
  $D_i(n) \leftarrow cwnd_i \times PL\ \forall i \in L_{sch}$
  $M \leftarrow |L_{sch}|$
  if $n = 1$ **then**
    $Q(0) \leftarrow \frac{R_{min} \times N_s \times T_s}{M}$
  end if
  $k \leftarrow \min_i\{RTT_i\}$
  $T \leftarrow kT_f$
  while $T > 0$ **do**
    $L_{active} \leftarrow \phi$
    for all $i \in L_{sch}$ **do**
      if $SNR_i(n) \geq SNR_{th}$ **then**
        $L_{active} \leftarrow L_{active} \cup \{i\}$
        $DC_i(n) \leftarrow DC_i(n-1) + Q(n-1)$
        $-R_i(n-1) \times N_i(n-1) \times T_s$
        if $Flag_{deadline}$ = 1 **then**
          $d_i(n) \leftarrow d_i(n-1)$
        else
          $d_i(n) \leftarrow 1$
        end if
      else
        $R_i(n) \leftarrow 0$
        $D_i(n) \leftarrow D_i(n-1)$
        $DC_i(n) \leftarrow DC_i(n-1) + Q(n-1)$
        if $Flag_{deadline}$ = 1 **then**
          $d_i(n) \leftarrow d_i(n-1) - T_f$
          if $d_i(n) \leq 0$ **then**
            $d_i(n) \leftarrow TTO_i$
          end if
        else
          $d_i(n) \leftarrow 1$
        end if
        $W_i(n) \leftarrow 0$
        $N_i(n) \leftarrow 0$
      end if
    end for
    for all $i \in L_{actve}$ **do**
      $D_i(n) \leftarrow D_i(n-1) - N_i(n-1) \times R_i(n-1) \times T_s$
      $dc_i(n) \leftarrow DC_i(n) + \min_j |DC_j(n)|, \forall j \in L_{active}$
      Map $R_i(n)$ to $SNR_i(n)$ in Table I
      $W_i(n) \leftarrow \frac{\frac{D_i(n)}{R_i(n)} \times \frac{dc_i(n)}{R_i(n)} / d_i(n)}{\sum_{j \in L_{active}} (\frac{D_j(n)}{R_j(n)} \times \frac{dc_j(n)}{R_j(n)} / d_j(n))}$
      $N_i(n) \leftarrow \frac{1}{T_s} \times \min\left(\frac{W_i(n) \times T_f}{\sum_{j \in L_{active}} W_j(n)}, \frac{D_i(n)}{R_i(n)}\right)$
      $Q(n) \leftarrow \frac{1}{M} \sum_{i \in L_{sch}} R_i(n-1) \times N_i(n-1) \times T_s$
    end for
    $T \leftarrow T - T_f$
    $n \leftarrow n + 1$
  end while
end while

TABLE VIII
AMOUNT OF DATA TRANSMITTED (Mb) (ADAPTIVE MODULATION)

	Equal Distances		Unequal Distances	
No.	TWUS-A	DTWUS-A	TWUS-A	DTWUS-A
1	67.89	68.46	77.30	77.21
2	67.65	68.24	68.14	69.28
3	67.90	68.98	57.50	58.67
4	67.90	68.34	77.12	76.37
5	67.92	68.43	72.57	73.60
6	67.74	68.51	58.12	58.22
7	67.87	68.41	68.05	68.80
8	67.92	68.15	68.09	69.10
9	67.80	68.41	57.68	58.90
10	67.99	68.41	67.39	67.44
Total	678.58	684.34	671.96	677.59

TABLE IX
JAIN'S FAIRNESS INDEX (JFI) AND USAGE

SS No.	Equal Distances		Unequal Distances	
Adaptive Modn.	TWUS-A	DTWUS-A	TWUS-A	DTWUS-A
JFI	0.999	0.999	0.989	0.990
% use	95.99	96.94	96.13	96.73
Fixed Modn.	TWUS	DTWUS	TWUS	DTWUS
JFI	0.999	0.999	0.987	0.990
% use	98.26	98.13	98.68	98.61

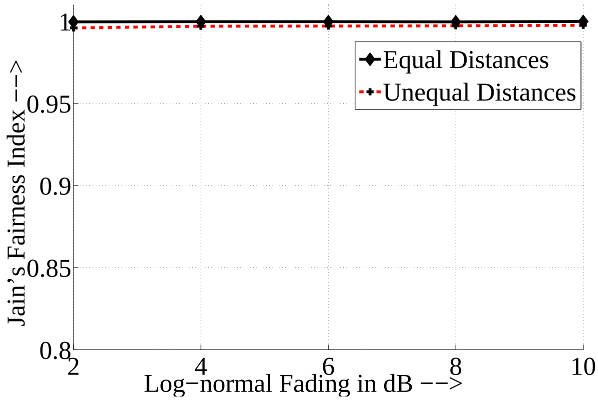

Fig. 4. JFI with Different log-normal Fading using TWUS-A

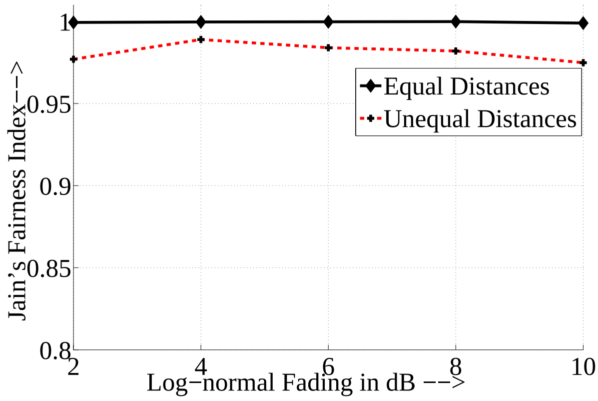

Fig. 5. JFI with Different Log-normal Fading using DTWUS-A