An Indexing Scheduler for Delay Constrainted Scheduling with Applications to IEEE 802.16

Nitin Salodkar
Department of Computer Science and Engineering,
Indian Institute of Technology Bombay,
Powai, Mumbai, India - 400076
Email: nitins@cse.iitb.ac.in

Abhay Karandikar
Department of Electrical Engineering,
Indian Institute of Technology Bombay,
Powai, Mumbai, India - 400076
Email: karandi@ee.iitb.ac.in

***Abstract*— We consider the problem of scheduling users on the downlink of a Time Division Multiplexed (TDM) system with constraints on the average packet delays over a fading wireless channel. Our objective is to maximize the sum throughput with constraints on the user delays. Due to the difficulty in computing optimal policy, we propose a suboptimal scheduling algorithm which is based on computing appropriate indices and scheduling the user with the highest index. Our simulations for the IEEE 802.16 system indicate that our algorithm satisfies the delay constraints of the users and is highly throughput efficient.**

I. Introduction

Wireless users perceive time varying channel quality. The channel quality across users might be quite diverse. In recent times, *cross layer* schemes that exploit the channel related information at the higher layers have resulted in improved system performance [1]. A centralized scheduler in a point-to-multipoint scenario, e.g., a base station on the downlink, can exploit this information from the physical layer by scheduling a user perceiving better channel quality. Data can be transmitted at a higher rate to such a user while maintaining an acceptable Bit Error Ratio (BER) at the receiver. This results in higher sum throughput [2] (i.e., the sum of the throughputs of all the users). Such schemes that exploit the opportunities provided by multiuser diversity for scheduling at the MAC layer are called *opportunistic* scheduling schemes [3]. However, under opportunistic scheduling over a Time Division Multiplexed (TDM) system, users perceiving perennially better channel conditions obtain a higher proportion of slots. In a best effort system, this leads to *unfairness*, while in a system providing QoS, this leads to QoS guarantees like average rate/delay guarantees being violated. Scheduling users perceiving poor channel conditions results in transmission at lower rates to such users, thereby reducing the sum throughput. The objective, therefore, is to schedule an appropriate user so as to maximize the sum throughput while satisfying the QoS constraints or being fair. Various fair scheduling algorithms have been explored in [3], [4], [5], [6].

A scheduling policy is considered *stable* if the expected queue length is bounded under the policy. In [7], the authors determine the throughput capacity region of a multi-access system, i.e., the set of all rates that can be delivered *reliably* under average power constraints. In [8], the authors have shown that the throughput capacity region is same as the multi-access stability region (i.e., the set of all arrival vectors for which there exists some rate and power allocation policies that keep the system stable). A scheduler is termed *throughput-optimal* if it can maintain the stability of the system as long as the arrival rate is within the stability region [9]. Throughput optimal scheduling policies have been explored in [7], [10]. Longest Connected Queue (LCQ) [11] , Exponential (EXP) [12], Longest Weighted Queue Highest Possible Rate (LWQHPR) [13] and Modified Longest Weighted Delay First (M-LWDF) [14] are other well known throughput optimal scheduling policies. It has been shown that the Longest Queue Highest Possible Rate (LQHPR) policy [15] (besides being throughput optimal) also minimizes the delay for any symmetric power control under symmetric fading provided that the packet arrival process is Poisson and packet length is exponentially distributed. Recently in [16], the author has studied the problem of minimizing sum power on the downlink under the user queue stability constraints. Note, however, that on the downlink, the base station typically transmits at a fixed maximum power sufficient to reach the farthest user and hence power minimization is not a central issue.

In this paper, we consider the problem of scheduling the users on the downlink of a TDM system such that the average packet delays experienced by the users are below certain specified constraints. For a multiuser queuing system with scheduler on a TDM channel, there is an extensive literature that we have reviewed above. However, the specific optimization problem of maximizing the sum throughput subject to constraints on the individual user delays has not been explicitly addressed so far. It can be easily argued that this problem has the structure of a Constrained Markov Decision Problem (CMDP) [17]. However, the primary difficulty in computing optimal policy (as exemplified later in this paper) lies in large state space size that increases exponentially with number of users. Moreover, computation of such a policy requires the knowledge of the system model, i.e., the knowledge of the probability distributions of the channel state and the arrival process for each user. This knowledge of the system model is not available in practice.

We believe that state space explosion and unknown system model are the primary reasons for inadequate attention towards

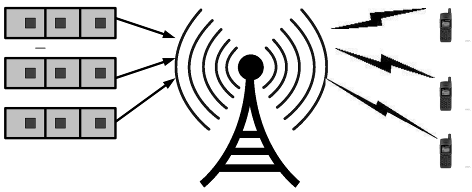

Fig. 1. System Model

optimal delay constrained multiuser scheduler structure despite abundant literature in wireless scheduling with various other performance objectives. We address this problem by proposing a suboptimal scheduler that is based on computing appropriate indices and scheduling the user with the highest index. The scheme generates indices in each slot in such a fashion that the delay constraints of the users are satisfied while still achieving a very high sum throughput. We demonstrate the applicability of our algorithm to IEEE 802.16 based system through simulation experiments. To the best of our knowledge, there is no scheme that solves the problem explicitly. For comparison purposes, we adapt the M-LWDF scheduler to our scenario just to illustrate that our algorithm achieves a high sum throughput even while satisfying the delay constraints.

II. System Model and Problem Formulation

A. System Model

We consider a multiuser TDM system where a base station schedules N users on the downlink, as depicted in Figure 1. Time is divided into slots of unit duration. Only one user can be scheduled in a time slot. The base station maintains a queue of size B bits for each user. Packets arrive into the queue at the beginning of a time slot. The packets are queued at the base station until they are transmitted. We assume that the number of packets arriving to the queue of user i in each slot forms an independent and identically distributed (i.i.d.) process. Let A_n^i denote the number of packets arriving into the user i queue in slot n. We assume that the arrivals in a slot are discrete and finite, i.e., the random variable A_n^i takes values from a set $\mathcal{A} \triangleq \{0, \ldots, A\}$. We assume that the distribution of A_n^i is not known.

The channel quality perceived by a user remains constant for the duration of a slot and changes from slot to slot in an i.i.d. fashion. This channel model is called block fading model [18]. Let y_n denote the signal transmitted by the base station in slot n. Then the signal R_n^i received by a user i in time slot n can be expressed as,

$$R_n^i = H_n^i y_n + G_n^i, \tag{1}$$

where H_n^i is a circularly symmetric complex Gaussian random variable and G_n^i is the Additive White Gaussian Noise (AWGN) at the receiver with Power Spectral Density (PSD) N_0. We define $X_n^i \triangleq |H_n^i|^2$ as the *channel state* for a user i in slot n. In practice, H_n^i is a continuous random variable and so is X_n^i. However, in this paper we assume that X_n^i takes values from a discrete and finite set $\mathcal{X}$. We assume perfect channel state information at the base station, i.e., in every slot, the base station has the perfect knowledge of the channel state perceived by each user.

Let Q_n^i denote the instantaneous queue length of a user i in slot n. Let U_n^i denote the number of bits that the base station can transmit *reliably* to user i in slot n. Since slot duration is normalized to 1, U_n^i also denotes the rate at which the base station can transmit to user i in slot n. Since the base station can at most transmit all the bits in a buffer in any slot, $U_n^i \leq Q_n^i$. We assume that U_n^i takes values from a finite and discrete set $\mathcal{U}$. The queue evolution equation for user i can be written as,

$$Q_{n+1}^i = \max(0, Q_n^i + A_{n+1}^i - I_n^i U_n^i), \tag{2}$$

where I_n^i is an indicator variable that is set to 1 if user i is scheduled in time slot n, otherwise it is set to 0. We assume that the buffer size B allocated for each user is large and that the probability of packet drops is negligible. We, therefore, ignore packet drops in (2). The users specify their QoS requirements in terms of average packet delay constraints. These constraints are known a priori to the base station.

B. Formulation as a Constrained Optimization Problem

By Little's law, the average delay $\bar{D}$ suffered by the packets is related to the average queue length $\bar{Q}$ as follows,

$$\bar{Q} = \bar{a}\bar{D}, \tag{3}$$

where $\bar{a}$ is the average arrival rate. For a constant average arrival rate, the average delay can be considered equivalent to the average queue length. Hence we consider constraints on average queue lengths instead of average delays in this paper. The long term average queue length for user i can be expressed as,

$$\bar{Q}^i = \limsup_{M \to \infty} \frac{1}{M} \sum_{n=1}^{M} Q_n^i. \tag{4}$$

Let $\bar{\mathbf{Q}} = [\bar{Q}^1, \ldots, \bar{Q}^N]^T$ denote the vector of long term average queue lengths. Let $\bar{\mathbf{a}} = [\bar{a}^1, \ldots, \bar{a}^N]^T$ denote the vector of the average arrival rates, $\bar{a}_i$ being the average arrival rate for user i. Let $\bar{\boldsymbol{\delta}} = [\bar{\delta}^1, \ldots, \bar{\delta}^N]^T$ denote the vector of queue length constraints. The sum throughput over a long period of time can be expressed as,

$$\bar{T} = \liminf_{M \to \infty} \frac{1}{M} \sum_{n=1}^{M} \sum_{i=1}^{N} I_n^i U_n^i. \tag{5}$$

The objective of the system is to maximize the sum throughput while satisfying the average packet delay requirements of the users. The scheduling problem can therefore be expressed as a constrained optimization problem,

$$\text{Maximize } \bar{T} \text{ subject to } \bar{Q}^i \leq \bar{\delta}^i \text{ for } i = 1, \ldots, N. \tag{6}$$

C. The CMDP Framework

Let $\mathbf{X}_n = [X_n^1, \ldots, X_n^N]^T$ denote the vector of channel states of the users in slot n. Let $\mathbf{Q}_n = [Q_n^1, \ldots, Q_n^N]^T$ denote the vector of queue lengths of the users in slot n. The state of the system in slot n is specified by the tuple $\mathbf{S}_n \triangleq (\mathbf{Q}_n, \mathbf{X}_n)$. The system state space $\mathcal{S} = \mathcal{Q}^N \times \mathcal{X}^N$ is discrete and finite. In each slot, the scheduler chooses a particular user based on the system state. The state of the system in the current slot depends on its state in the previous slot and the decision taken by the scheduler in the previous slot. The objective is to determine an optimal policy that achieves the highest possible throughput while satisfying the delay constraints of the users. Hence the problem has the structure of a CMDP [17]. However, the traditional approaches based on Linear Programming (LP) [17] for determining the optimal policy cannot be used because of the following reasons:

1) *Large state space:* In our model, the system state space is large even for moderate number of users. We illustrate this with a simple example. Consider a system with 4 users. Assume that the base station reserves a buffer of size 50 packets (assuming equal sized packets) for each user. Assume that the channel condition of each user can be represented using 8 states, which is a practical assumption justified in [19]. For this scenario, the system state space contains $50^4 \times 8^4 = 2.56 \times 10^{10}$ states. The computational complexity of the traditional LP based approaches is proportional to the state space size [17] and hence the computational complexity also increases exponentially with users.
2) *Unknown system model:* Traditional approaches require the information about the state transition probability of the underlying Markov process which in turn needs the information regarding the probability distributions of the channel state and the arrival for each user. This information is not available in practice.

To alleviate the problem of unknown system model, reinforcement learning algorithms [20] could be used. However, with such a large state space, the learning algorithms would take prohibitively large time to converge to the optimal scheduling policy. Hence, we develop an indexing scheme which though suboptimal, yet performs very well and does not face these problems.

III. Indexing Scheduler

We propose an *indexing* scheme that generates indices for all users in each slot. We seek to generate indices that exploit the tradeoff between maximizing throughput and satisfying delay constraints. The user having the maximum index in a slot is scheduled in that slot. Note that maximizing the sum throughput requires that a user with the best channel state be scheduled in a slot. On the other hand, if the queue length of a user exceeds the queue length constraint, then the scheduler has to compromise on the objective of maximizing the sum throughput and possibly schedule a user not having the best channel state. These considerations can be precisely expressed as follows:

1) To fulfill the objective of maximizing the sum throughput, an index must be proportional to the channel state of the user. This ensures that a user with a better channel state has a higher probability of being scheduled.
2) The index allocation must be cognizant of the user delay requirements. A user having a tighter delay constraint must be given a higher index and hence higher probability of being scheduled. If the slots allocated to a user are not sufficient to satisfy its delay constraint, its queue length would be greater than the desired queue length frequently. In order to satisfy the delay constraint of a user, its index must be proportional to the aggregate amount with which its queue length exceeds the desired queue length. This ensures that the user having a higher aggregate queue constraint violation has a greater probability of being scheduled.

Taking these requirements into consideration, we define the index κ_n^i of a user i in a slot n as:

$$\kappa_n^i = \lambda_n^i \times U_n^i. \quad (7)$$

λ_n^i is the weight of user i in slot n. This weight is dynamically adjusted in each slot based on the deviation of the queue length of that user from its desired queue length. Once the indices are determined, the algorithm determines the user with the highest index with a non-empty queue and non-zero rate, and schedules this user. If there are multiple such users, one of them is scheduled randomly with uniform probability. We now describe an approach for determining the weight λ_n^i for a user i in slot n.

A. Determining the Weights

As outlined above, if the aggregate queue constraint violation of a user is large, it must have a large weight. Hence, we dynamically update the weight in each slot by adding the deviation of the current queue length from the constraint to it. Consider sequence $\{a_n\}$ that satisfies the following properties:

$$\lim_{n\to\infty} a_n = 0, \quad \lim_{n\to\infty} \sum_n (a_n)^2 < \infty, \quad \lim_{n\to\infty} \sum_n a_n = \infty. \quad (8)$$

The first two properties in (8) ensure that the sequence $\{a_n\}$ converges to zero sufficiently rapidly, while the third property ensures that it does not converge to zero too rapidly. Let $\lambda_0^i = 1 \; \forall i$. The weight λ_n^i for a user i in slot n is then determined using the following iteration:

$$\lambda_{n+1}^i \quad = \quad \min(L, \max(0, \lambda_n^i + a_n \times (Q_n^i - \bar{\delta}^i))), \quad (9)$$

where $L >> 0$, i.e., we project the λ^i iterates in the interval $[0, L]$. The properties of $\{a_n\}$ ensure that the update rate of weight λ^i is neither too fast nor too slow (following arguments similar to that of [21]). The stable value of the weight determines the proportion of slots allocated to a user based on its delay requirement and thereby the relative priority between the users. The intuition behind (9) is to iteratively tune the weight of user i so as to satisfy its delay constraint. If

Q_n^i continues to be less than $\bar{\delta}^i$ then it progressively reduces the weight λ_n^i in the subsequent slots thereby reducing the probability of user i being scheduled. On the other hand, if Q_n^i continues to be more than $\bar{\delta}^i$, then it progressively increases the weight λ_n^i thereby increasing the probability of user i being scheduled and hence increasing the proportion of slots that would be allocated to user i. Updation of the weights results in a redistribution of the proportion of slots allocated to users. If the delays are feasible, the scheme determines an allocation such that the delay constraints of all the users are satisfied.

Theorem 1: $\boldsymbol{\lambda}_n$ iterates converge to a stable value $\boldsymbol{\lambda}_*$.

Proof: We provide a sketch of the proof in the Appendix. ■

IV. Simulation Setup and Results

In this section, we demonstrate the following:

1) The algorithm satisfies the delay constraints of all the users.
2) The algorithm is efficient in terms of the achieved sum throughput through comparison with the M-LWDF [14] scheduler.

M-LWDF scheme considers the probability with which a user's queue length is allowed to exceed a certain target queue length. We assume that this probability is the same for all the users and ignore it in the present simulations. Specifically, the adapted M-LWDF schedules a user i in each slot such that,

$$i = \arg\max_j \tau_n^j \times U_n^j, \tag{10}$$

where τ_n^j is the delay experienced by the head of the line packet for user j. M-LWDF scheme transmits at a constant power in each time slot. Note that M-LWDF scheduler attempts to minimize the user delays and does not address the problem of maximizing sum throughput subject to delay constraints. We, therefore, first determine the average delays experienced by the users under the M-LWDF scheme for various average arrival rates. The values of these delays are then considered to be the delay constraints for the indexing scheduler. We determine the average delays experienced by the users under the indexing scheduler and also the sum throughput achieved under it. We perform the simulations within the framework of an IEEE 802.16 system.

A. The IEEE 802.16 System

In this paper, we concentrate on the point-to-multipoint (PMP) mode specified in the IEEE 802.16 standard, where a centralized base station (BS) serves multiple subscriber stations (SSs). We consider the downlink (DL) transmissions in the *residential* scenario as in [22] where the BS provides Internet access to the subscribers. IEEE 802.16 medium access control (MAC) specifies the non real-time polling service (nrtPS) for non real-time applications. Although the standard does not explicitly specify any mechanism for providing average delay guarantees, the nrtPS service discipline can be extended to cater to the average delay requirements of the users. The unicast polling service of nrtPS can be used to determine the channel state perceived by the users. On the downlink, the base station has the knowledge of the queue lengths of all the users. The scheduling algorithm can thus be implemented as a part of nrtPS.

The system can be operated in either time division duplex (TDD) or frequency division duplex (FDD) mode. We assume the FDD mode of operation where all SSs have full-duplex capability. We consider a single carrier system with a frame duration of 1 msec and bandwidth of 10 MHz. The SSs employ the following modulations: 64-Quadrature Amplitude Modulation (QAM), 16-QAM, Quadrature Phase Shift Keying (QPSK) and QPSK with a rate 1/2 code; along with a filter rolloff factor of 0.22. These provide us with the following 4 rates of transmission: 24 Mbps, 16 Mbps, 8 Mbps and 4 Mbps respectively. We consider 20 connections on the DL and assume that the number of connections does not change over the duration of the simulations. We measure the sum of queuing and transmission delays of the packets and ignore the propagation delays.

B. Simulation Details and Results

Internet traffic is modeled as a web traffic source [23]. Packet sizes are drawn from a truncated Pareto distribution (shape factor 1.2, mode = 2000 bits, cutoff threshold = 10000 bits) which provides us with an average packet size of 3860 bits. In each time frame, we generate the arrivals for all the users using Poisson distribution. Arrivals are generated in an i.i.d. manner across frames.

We simulate a Rayleigh channel[1] for each user. For a Rayleigh model, channel state X^i is an exponentially distributed random variable with probability density function given by $f_{X^i}(x) = \frac{1}{\alpha^i} e^{-\frac{x}{\alpha^i}}$, where α^i is the mean of X^i. We discretize the channel into eight equal probability bins with the boundaries specified by $\{$ (-∞, -8.47 dB), [-8.47 dB, -5.41 dB), [-5.41 dB, -3.28 dB), [-3.28 dB, -1.59 dB), [-1.5 dB, -0.08 dB), [-0.08 dB, 1.42 dB), [1.42 dB, 3.18 dB), [3.18 dB, ∞) $\}$. We associate a channel state with each bin. The channel state space is $\mathcal{X} = \{-13$ dB, -8.47 dB, -5.41 dB, -3.28 dB, -1.59 dB, -0.08 dB, 1.42 dB, 3.18 dB$\}$. This discretization of the state space of X^i has been justified in [24]. Users are divided into two groups (Group 1 and Group 2) of 10 users each. In all the scenarios described below, each simulation run consists of simulating the algorithms for 100000 frames. Results are presented after averaging over 20 simulation runs.

Scenario 1: In this scenario, we demonstrate that the algorithm satisfies the various user specified delay constraints. We consider two cases: symmetric case and asymmetric case. In both the cases, in each frame, arrivals are generated with a Poisson distribution with mean 180 packets/sec/user. Packet lengths are Pareto distributed with shape factor 1.2, mode = 2000 bits and cutoff threshold = 10000 bits. This results in an average arrival rate of 0.6948 Mbits/sec/user. We choose

[1]The scheduling algorithm is not aware that the channel is Rayleigh or that the arrival distribution is Poisson.

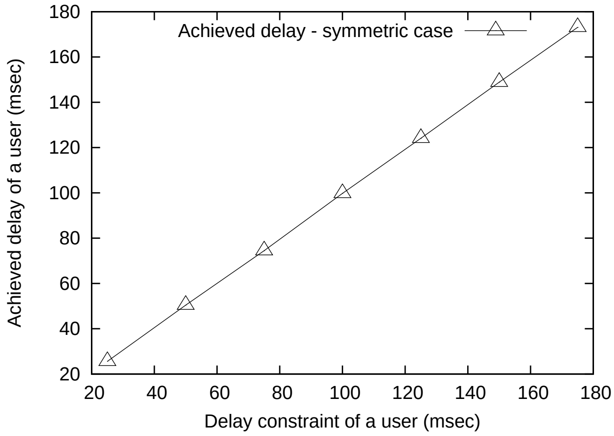

Fig. 2. Delay experienced by a user selected at random - symmetric case

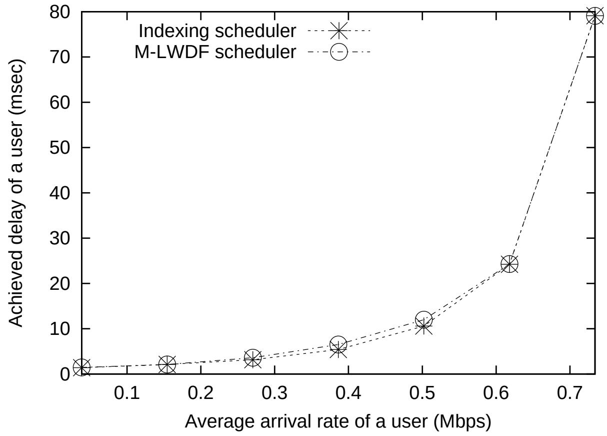

Fig. 4. Delays experienced under Indexing scheduler with those under M-LWDF scheduler as constraints

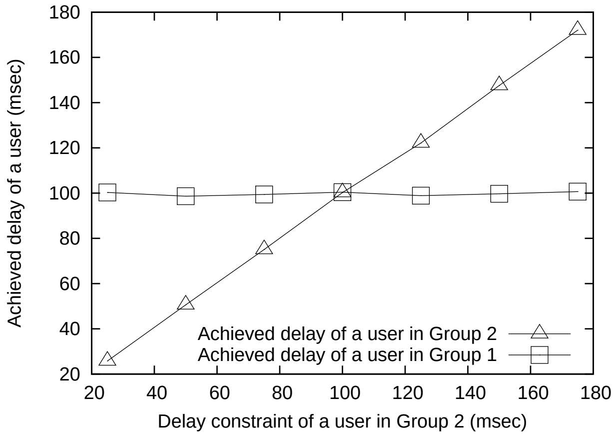

Fig. 3. Delay experienced by two users selected at random from Group 1 and Group 2 - asymmetric case

$\alpha^i = 0.4698\ (-3.28\text{ dB})\ \forall i$. In each frame, we generate X^i using exponential distribution with mean α^i. We determine the channel state based on the channel bin that contains X^i as explained above. We fix the transmission power at 4 Watts. The indexing algorithm determines the user that is scheduled in that frame. In the symmetric case, we fix the delay constraints of all the users to be equal to 25 msec. We then repeat the experiment with different values of delay constraint such as $50, 75, 100, 125, 150, 175$ msec and measure the average delay experienced by each user. The value of the delays for a particular user (chosen at random) are plotted in Figure 2. In the asymmetric scenario, the delay constraint of the users in Group 1 is fixed at 100 msec for all the experiments, while the delay constraint of the users in Group 2 is varied as $25, 50, 75, 100, 125, 150, 175$ msec in successive experiments. The average delay experienced by two specific users (each selected at random from Group 1 and Group 2) are plotted in Figure 3. It can be seen from Figures 2 and 3 that the delay constraints are satisfied in both the cases.

Scenario 2: In this scenario, we demonstrate that the system achieves a high sum throughput. We first simulate the M-LWDF scheme. In each frame, arrivals are generated with a Poisson distribution. In successive experiments, the mean arrival rate is fixed at $10, 40, 70, 100, 130, 160, 190$ packets/sec/user respectively. Packet lengths are Pareto distributed with shape factor 1.2, mode = 2000 bits and cutoff threshold = 10000 bits. This results in an average arrival rate of 0.0386 to 0.7334 Mbits/sec/user in successive experiments. Rest of the parameters are same as in Scenario 1. We determine the delays experienced by the users and the sum throughput achieved. The delays experienced by the users in the M-LWDF scheme serve as delay constraints for the users in the indexing scheduler. We determine the delays experienced by a particular user selected at random and the sum throughput achieved under the indexing scheduler and compare these with those of the M-LWDF scheme in Figures 4 and 5 respectively. From Figure 4 it can be seen that the delays experienced by a user under the indexing scheduler are *less than or equal to* those under the M-LWDF scheme implying that the delay constraints are satisfied. Moreover, from Figure 5, it can be seen that the sum throughput achieved by the indexing scheduler is very close to that achieved by the M-LWDF scheme. Note that M-LWDF algorithm attempts to minimize the delay and does not address the problem considered in this paper (i.e., of maximizing the sum throughput subject to satisfying the user delays). The indexing scheduler, on the other hand, caters to the delay constraints, and while doing it, we have demonstrated through simulations that it also achieves a high sum throughput.

V. Conclusion

In this paper, we have considered the problem of scheduling users on the downlink of a TDM system with constraints on the average packet delays over a fading wireless channel. For a multiuser queuing system with scheduler on a TDM channel, there is an extensive literature on scheduling algorithms. However, the specific optimization problem of maximizing the sum throughput subject to constraints on the individual user delays has not been explicitly addressed so far. This is due to the fact that while this problem has the structure of a CMDP; the large system state space and unknown system model render the

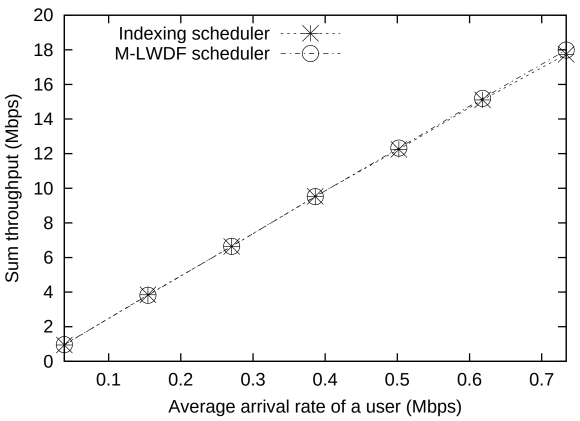

Fig. 5. Sum throughput

traditional approaches for determining the optimal policy for a CMDP infeasible. Hence, we have suggested a suboptimal indexing scheduler that is easy to implement in practice. The indexing scheduler generates indices in each time slot and the user with the maximum index is scheduled. Our simulations for the IEEE 802.16 system have indicated that the delay constraints of the users are satisfied while still achieving a high sum throughput. While we have simulated the algorithm within the IEEE 802.16 framework, it is applicable to any TDM based wireless downlink scheduling framework.

APPENDIX

Sketch of Proof of Theorem 1: Let $\boldsymbol{\lambda}_n = [\lambda_n^1, \ldots, \lambda_n^N]^T$ denote the weight vector in slot n. (9) can be expressed in the vector form as,

$$\boldsymbol{\lambda}_{n+1} = \min(\mathbf{L}, \max(\mathbf{0}, \boldsymbol{\lambda}_n + a_n \times (\mathbf{Q}_n - \bar{\delta}))). \quad (11)$$

We consider the $\boldsymbol{\lambda}$ and $\mathbf{Q}$ values after T slots for large T. Let b_l denote the value of b at the $(T \times l)$ th slot. Note that if the weight λ^i of user i is increased, over a period of time, its queue length Q^i reduces, thus increasing $\bar{\delta}^i - Q^i$. We model this effect using the following equation:

$$\mathbf{Q}_{l+1} - \boldsymbol{\delta} = G(\boldsymbol{\lambda}_l) \implies \mathbf{Q}_{l+1} = \bar{\boldsymbol{\delta}} + G(\boldsymbol{\lambda}_l), \quad (12)$$

where $G(\cdot)$ is a monotonically non-increasing continuous function of $\boldsymbol{\lambda}$. Moreover, if $Q^i - \bar{\delta}^i$ increases, λ^i increases. We model this effect using the following equation:

$$\boldsymbol{\lambda}_{l+1} = F(\mathbf{Q}_l - \bar{\boldsymbol{\delta}}), \quad (13)$$

where $F(\cdot)$ is a monotonically non-decreasing continuous function of $\mathbf{Q}_l - \bar{\boldsymbol{\delta}}$. (12) and (13) form the following fixed point iteration:

$$\boldsymbol{\lambda} = F(G(\boldsymbol{\lambda})). \quad (14)$$

Since $F(G(\cdot))$ is a composition of continuous functions, it is continuous. Thus we have a continuous mapping from $[0, L]$ to $[0, L]$. Hence, by Brouwer's fixed point theorem [25], there exists a fixed point in $[0, L]$. $G(\boldsymbol{\lambda})$ being non-increasing in $\boldsymbol{\lambda}$, $F(G(\boldsymbol{\lambda}))$ is non-increasing in $\boldsymbol{\lambda}$. Hence, the fixed point is unique, say $\boldsymbol{\lambda}_*$ which is denoted as the stable value. The detailed proof is omitted do to space constraint.

REFERENCES

[1] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, 2003.

[2] R. Knopp and P. A. Humblet, "Information Capacity and Power Control in Single-Cell Multiuser Communications," in *Proceedings of IEEE ICC*, vol. 1, Seattle, USA, June 1995, pp. 331–335.

[3] X. Liu, E. Chong, and N. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *IEEE Journal on Selected Areas in Communication*, vol. 19, no. 10, pp. 2053–2064, 2001.

[4] F. Berggren and R. Jantti, "Asymptotically Fair Transmission Scheduling over Fading Channels," *IEEE Transactions on Wireless Communication*, vol. 3, no. 1, pp. 326–336, Jan. 2004.

[5] S. Kulkarni and C. Rosenberg, "Opportunistic Scheduling Policies for Wireless Systems with Short Term Fairness Constraints," in *Proceedings of IEEE GLOBECOM*, vol. 1, Dec. 2003, pp. 533–537.

[6] Y. Liu and E. Knightly, "Opportunistic Fair Scheduling over Multiple Wireless Channels," in *Proceedings of IEEE INFOCOM*, vol. 2, San Francisco, USA, Mar. 2003, pp. 1106–1115.

[7] D. Tse and S. Hanly, "Multi-access Fadaing Channels - Part I: Polymatroid Structure, Optimal Resource Allocation and Throughput Capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, 1998.

[8] E. Yeh and A. Cohen, "Throughput and Delay Optimal Resource Allocation in Multiaccess Fading Channels," in *Proceedings of International Symposium on Information Theory*, 2003, p. 245.

[9] R. Berry and E. M. Yeh, "Cross-Layer Wireless Resource Allocation," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 59–68, 2004.

[10] M. J. Neely, E. Modiano, and C. Rohrs, "Power and Server Allocation in Multi-beam Satellite with Time Varying Channels," in *Proceedings of IEEE INFOCOM*, San Francisco, USA, 2002, pp. 138–152.

[11] L. Tassiulas and A. Ephremides, "Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity," *IEEE Transactions on Information Theory*, no. 2, pp. 466–478, 1993.

[12] S. Shakkottai and A. Stolyar, "Scheduling for Multiple Fows Sharing a Time-varying Channel: the Exponential Rule," *AMS Translations Series 2*, vol. 207, 2002.

[13] E. Yeh and A. Cohen, "Delay Optimal Rate Allocation in Multiaccess Fading Communications," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Oct. 2002, pp. 404–407.

[14] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing Quality of Service over a Shared Wireless Link," *IEEE Communications Magazine*, vol. 39, Feb. 1996.

[15] E. Yeh, "Multiaccess and Fading in Communication Networks," PhD Thesis, Massachusetts Institute of Technology, September 2001.

[16] M. J. Neely, "Optimal Energy and Delay Tradeoffs for Multi-user Wireless Downlinks," in *Proceedings of IEEE INFOCOM*, 2006, pp. 1–13.

[17] E. Altman, *Constrained Markov Decision Processes*. Boca Raton, FL: Chapman and Hall/CRC Press, 1999.

[18] R. A. Berry and R. G. Gallager, "Communication over Fading Channels with Delay Constraints," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1135–1149, 2002.

[19] H. S. Wang and N. Moayeri, "Finite-State Markov Channel – A Useful Model for Radio Communication Channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, Feb. 1995.

[20] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[21] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[22] C. Cicconeti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of Service Support in IEEE 802.16 Networks," *IEEE Network*, pp. 50–55, March/April 2006.

[23] T. G. Neame, M. Zukerman, and R. G. Addie, "Modeling Broadband Traffic Streams," in *Proceedings of IEEE GLOBECOM*, 1999, pp. 1048–1052.

[24] H. Wang and N. Mandayam, "A Simple Packet Transmission Scheme for Wireless Data over Fading Channels," *IEEE Transcations on Communications*, vol. 52, no. 7, pp. 1055–1059, 2004.

[25] R. K. Sundaram, *A First Course in Optimization Theory*. Cambridge University Press, 1996.