Power efficient broadcast scheduling with delay deadlines

Dinesh Rajan
Department of Electrical Engineering
Southern Methodist University
Dallas, Texas
rajand@engr.smu.edu

Ashutosh Sabharwal, Behnaam Aazhang
Department of Electrical and Computer Engineering
Rice University
Houston, TX
{ashu,aaz}@rice.edu

Abstract

In this paper, we present a framework for the design of minimal power schedulers that satisfy average packet delay bounds for multiple users in a Gaussian wireless broadcast channel. We completely characterize the achievable region in the multidimensional delay-power space, and present schedulers that achieve the boundary regions. The optimal schedulers minimize the transmission power by jointly allocating rate and power to the different users based on various buffer and channel conditions. Finally, we also present low complexity scheduler designs that have near optimal performance.

1. Introduction

Scheduling is an important technique in achieving higher throughputs or lower powers in wireless networks [1–3, 6, 10–12]. We showed that in the single user scenario, transmission power can be significantly reduced by increasing packet delays in both fading channels and additive white Gaussian noise (AWGN) channels with bursty input [15, 16]; this was also independently reported in [4, 7, 13]. In AWGN channels, when the input traffic is bursty, reduction in transmission power with increasing delays was demonstrated by smoothening the input traffic [15] and transmitting nearly constant rate traffic (subject to meeting the delay and buffer constraints). In this sequel, we explore power efficient scheduling for multiple users in a Gaussian broadcast channel with average packet delay constraints.

Consider a system in which packets intended for multiple users arrive at a single transmitter and are stored in separate buffers. A scheduler is defined as a mapping from the vector of buffer states corresponding to the different users, to the vector of instantaneous transmission rates for each user. We investigate scheduling in a class of *randomized* mappings (defined in Sec. 2) and show that the average delay experienced by the users and the total transmission power using these randomized mappings can be expressed as convex combinations of the average delays and power, respectively, using *deterministic* mappings. As a consequence we show that the achievable region (defined in Sec. 2) in the multidimensional delay-power space is convex; moreover, we also show that the region is piecewise planar. This achievable region is analogous to the delay limited capacity region in multiple access fading channels that has a polymatroidal structure [9]. We also present average delay bounded multiuser schedulers, that achieve the boundary of the achievable region, *i.e.*, satisfies the different users delay constraints with minimal power consumption. The optimal schedulers, which are computed using a dynamic program, vary the rate and power for each user based on the joint buffer and channel states of all the users.

The complexity of the dynamic program used to calculate optimal schedulers is exponentially high for a system with large number of users and large buffer sizes. We propose a set of low complexity schedulers, which are shown to have near optimal delay-power performance. Finally, we derive a closed form approximation for multiuser power-delay relationship in a broadcast system. This approximation gives a coarse estimate of the power needed to support a large set of users simultaneously with different delay constraints. The results presented in this paper are useful in system planning, resource allocation and to make call admission control decisions. Even though power optimization is more critical in mobile handsets, minimizing the transmission power at the base station to achieve the desired in-cell performance results in lowering the intercell interference, thereby improving overall system throughput. The power optimization at the transmitter is also useful in other scenarios like ad hoc networks in which mobile nodes could broadcast information to multiple nodes simultaneously.

In this paper, we make some simplifying assumptions to derive the optimal schedulers. We use an *i.i.d.* arrival model for the source traffic and use average packet delay as a measure of quality of service provided. Although there exists more sophisticated models which represent realistic sources more accurately, we found that the analysis was nontrivial even using these simplified models. Thus, we hope to create a foundation for power efficient broadcast scheduling using these simple models and capture the multiuser effects in an analytical manner. For simplicity, the analysis in this paper is given for AWGN channels using capacity achieving broadcast transmission. However, all the results presented can be extended to Markovian fading channels and TDMA systems in a straightforward manner; see [14] for details.

The remainder of this paper is organized as follows. In Sec. 2, we introduce some basic notation and formalize the scheduling problem of interest. The achievable region is computed and optimal schedulers are presented in Sec. 3. Low complexity scheduler design is considered in Sec. 4. Finally, we conclude in Sec. 5.

2. Problem Formulation

We consider a time-slotted system with K flows[1] being transmitted from a single transmitter to K different receivers over a shared wireless downlink channel. The flows are arbitrarily labeled from 1 to K. In the n^{th} time-slot, flow i receives $a_{i,n} \in \{0, 1, 2, \ldots, M_i\}$ equal size packets at an average rate λ_i. Further, the arrival process of each flow, $a_{i,n}$, is assumed to be independent of other flows, and independent from one time-slot to another. Let $1 \times K$ vector $\mathbf{a}_n = \begin{bmatrix} a_{1,n} & a_{2,n} & \cdots & a_{K,n} \end{bmatrix}$. Throughout this paper, we represent vector quantities in **boldface**. We assume that each flow's packets are stored in a separate buffer of size L_i for flow i and the maximum arrivals in any slot, M_i, is less than L_i. The number of queued packets at the beginning of the n^{th} time-slot in buffer i is denoted by $x_{i,n}$; the set of all buffer states are represented by $1 \times K$ vector $\mathbf{x}_n = \begin{bmatrix} x_{1,n} & x_{2,n} & \cdots & x_{K,n} \end{bmatrix}$. A schematic of system is given in Fig. 1. The transmitted signal X_n in the n^{th} time-slot is given by

$$X_n = \sum_{i=1}^{K} P_{i,n} S_{i,n}, \tag{1}$$

where $P_{i,n}$ is the power used to transmit the data $S_{i,n}$ for user i in the n^{th} time-slot. The data $S_{i,n}$ depends on the number of packets $u_{i,n}$ transmitted for user i and on the channel coding, modulation and waveform used. We propose to use information theoretically optimal coding and modulation to obtain $S_{i,n}$, which serves two important purposes. First, we can use a closed form relation between amount of data $u_{i,n}$ and the minimum required power $P_{i,n}$ to transmit it reliably (details in Sec. 2.1). Second, it gives us a universal lower bound on power required by any scheduler designed for a specific coding and modulation scheme. The total transmit power P_n during time-slot n equals $\sum_{i=1}^{K} P_{i,n}$ and is computed in (6). Let $\mathbf{u}_n = \begin{bmatrix} u_{1,n} & u_{2,n} & \cdots & u_{K,n} \end{bmatrix}$. The buffer update is given by,

$$x_{k,n+1} = \min\left(x_{k,n} + a_{k,n} - u_{k,n}, L_k\right), \quad \forall\ k, n. \tag{2}$$

The received signal $Y_{i,n}$ at the i^{th} mobile node is given by

$$Y_{i,n} = X_n + \eta_{i,n}, \tag{3}$$

where $\eta_{i,n}$ is the additive white Gaussian noise with zero mean and variance $N_i > 0$.[2] The receiver decodes the data based on $Y_{i,n}$ and its *a priori* knowledge of the coding scheme.

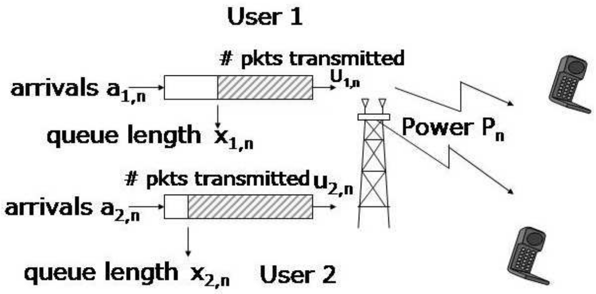

Figure 1. Schematic of Broadcast system.

The main objective of our scheduler design is to minimize the expected transmitted power among the class of randomized stationary zero-loss schedulers, Θ (defined in Sec. 2.2). Formally,

$$P^*(\mathbf{D}_o) = \min_{\substack{\Theta \\ \mathbb{E}\{x_{k,n}\} \le \lambda_k D_{k,o}, \forall k}} \lim_{n \to \infty} \mathbb{E}\left\{P_n\right\}, \tag{4}$$

where $D_{k,o}$ is the delay bound for the k^{th} user; define $\mathbf{D}_o = \begin{bmatrix} D_{1,o} & D_{2,o} & \cdots & D_{K,o} \end{bmatrix}$ as the $1 \times K$ vector of delay bounds.

Definition 1 (Achievability) *A point $(\mathbf{D}, P)$ in the $(K+1)-$dimensional delay-power space is achievable if there exists a scheduler that achieves delays no greater than $\mathbf{D}$ and average transmission power P with zero packet loss.*[3]

Given $P^*(\mathbf{D}_o)$, all points $(\mathbf{D}_o, P)$ such that $P \geq P^*(\mathbf{D}_o)$ are achievable. In Sec. 3, we compute the entire achievable region. In Sec. 2.1 we discuss the power control used and in Sec. 2.2 define some basic properties of the class of schedulers considered.

[1] In this paper, we use the terms users and flows interchangeably.

[2] Extensions of this model to block fading channels are direct [14].

[3] For two vectors $\mathbf{d}$, $\mathbf{D}$ we mean $\mathbf{d} \leq \mathbf{D}$ iff the inequality holds element wise.

2.1. Multiuser Power Control

In each time-slot, after choosing the number of packets $\mathbf{u}_n$ to be transmitted, the physical layer encoding selects a power level $P_{i,n}$ and coding for each user to ensure reliable reception. A coding and power control scheme is reliable if the received packets can be decoded with exceedingly small errors. Instead of relying on any specific coding scheme, we use code-independent information theoretic bounds giving us a simple closed form expression for power control. Though the information theoretic bounds rely on asymptotic analysis, state of the art coding schemes (like LDPC and Turbo codes) closely follow the predicted performance trends. Lastly, all the results in this section assume that each packet is reasonably long.

Note that the length of the time-slot is fixed in our system model. Thus, if the scheduler chooses more packets for transmission in any time-slot, the rate of transmission has to be proportionally scaled. Given a set of rates $\mathcal{R} = (R_{1,n}, R_{2,n}, \ldots, R_{K,n})$ that needs to be transmitted in a time-slot, the required power using optimal broadcast coding satisfies the following set of inequalities [8],

$$\begin{aligned} R_{k,n} &= u_{k,n}F < W \log\left(1 + \frac{P_{k,n}}{WN_k + \sum_{i=1}^{k-1} P_{i,n}}\right) \qquad (5)\\ &\forall k = 1, 2, \ldots, K \end{aligned}$$

where W represents the channel bandwidth and $F = \frac{S}{T_s}$ represents the normalization factor required to convert from the number of packets to rate (in bits per second) and depends on the time period of each time-slot T_s and size of each packet S. The minimum total power required (denoted by P_{bc}) is thus given by

$$P_{bc}(\mathbf{u}_n) = W \sum_{i=1}^{K} \left(e^{\frac{u_{i,n}F}{W}} - 1\right) N_i \prod_{j=i+1}^{K} e^{\frac{u_{j,n}F}{W}}. \qquad (6)$$

For simplicity, we assume $F = 1$ in the remainder of the paper.

The main observation from this section is that the total transmit power for each user depends exponentially on the number of packets $\mathbf{u}_n$ transmitted in a time-slot. Although based on an asymptotic analysis, any practical low-error multirate transmission scheme will have similar exponential relationship between power and information rate. This nonlinear (convex) relationship is the fundamental reason why delay can be traded for power even for single flow case without channel variations [13, 16]. The gains of scheduling without channel time-variations are also present in the multiple flow case as shown in the subsequent sections.

2.2. Admissible Schedulers

A scheduler α is a mapping from the buffer state $\mathbf{x}_n$ to the number of packets scheduled for transmission $\mathbf{u}_n$, $\alpha : \mathbf{x}_n \mapsto \mathbf{u}_n$. If the mapping α is deterministic, *i.e.*, the scheduler transmits the same number of packets for a given buffer state, then it is labeled as *deterministic* scheduler. We denote by $\mathbb{W}$ the set of all deterministic schedulers.

In contrast, a scheduler is *randomized* if the mapping α is probabilistic, $\alpha : p(\mathbf{u}_n|\mathbf{x}_n)$. A scheduler is considered *stationary* if the mapping is independent of time index n. The randomization of the scheduler serves an important purpose. If limited to the class of deterministic schedulers, the number of possible schedulers is finite and thus the set of achievable delay vectors $\mathbf{D}_o$ is finite. Thus, not all delay tuples can be achieved with equality using only deterministic schedulers. Since randomized schedulers subsume the class of deterministic schedulers, they always require less or equal power than their deterministic counterparts, and permit achieving arbitrary delay vector $\mathbf{D}_o$ if at all possible.

Denote by Ω the set of all possible states of the buffer, $\{\mathbf{x}_n^i\}_i$, where $\mathbf{x}_n^i$ is a particular value (state) of $\mathbf{x}_n$. The cardinality of the set Ω is $|\Omega| = \prod_{k=1}^{K}(L_k+1)$. For any given scheduler α denote by $C_{i,j}$ the probability of transition from state $\mathbf{x}^j \in \Omega$ to $\mathbf{x}^i \in \Omega$ (the time index n is suppressed since the schedulers are assumed to be stationary). Denote the stationary probability of being in state $\mathbf{x}^i$ by s_i. From the assumed stationarity of the schedulers, it follows that

$$\mathbb{C}\mathbf{s} = \mathbf{s}, \qquad (7)$$

where $\mathbb{C} = \left[C_{ij}\right]$, $\mathbf{s} = \left[s_i\right]$ $i, j = 1, \ldots, |\Omega|$.

Given a particular scheduler α, we can compute the resulting delays and required transmission power as follows. The average delay D_k for user k can be computed using Little's formula [5] and is given by

$$D_k = \frac{1}{\lambda_k} \sum_{\mathbf{x}^i \in \Omega} x_k^i s_i, \qquad (8)$$

where x_k^i is the number of packets in the k^{th} buffer when buffer state equals $\mathbf{x}^i$. The total average transmission power is given by

$$P_{avg} = \sum_{\mathbf{x}^i \in \Omega} \sum_{\mathbf{u}} s_i P_{bc}(\mathbf{u}) p(\mathbf{u}|\mathbf{x}^i), \qquad (9)$$

where $p(\mathbf{u}|\mathbf{x}^i)$ is the probability of transmitting $\mathbf{u}$ packets in state $\mathbf{x}^i$.[4]

Among the class of randomized stationary schedulers, we limit our attention to schedulers which do not drop any packets either intentionally or due to buffer overflow. Buffer

[4]The dependence of $\mathbb{C}$, $\mathbf{s}$, P_{avg}, D_k on scheduler α is not explicitly shown when it is clear from the context.

overflows can be prevented because the maximum burst size of the input traffic is assumed to be known. Formally,

$$\Theta = \{\alpha : \alpha \text{ is zero-loss }\}. \qquad (10)$$

A scheduler is considered zero-loss if it does not drop any packets, does not allow buffer overflow and uses adequate power to ensure that receivers can decode packets with arbitrarily small probability of error. Using the power as given by (6) ensures that all transmitted packets are transmitted with arbitrarily small probability of error. Note that the proposed formulation seeks the power-minimizing scheduler within the class Θ, but in power-constrained systems, the resultant scheduler may require more power than available. Thus, the case where more than available power is needed implies the power-delay pair is not implementable, thereby providing a method for admission control, rate throttling or relaxing delay guarantees. The following theorem characterizes the set Θ.

Theorem 1 (Zero Buffer Overflow) *Consider K flows with independent queues, each with finite buffers of size L_k. Flow k has a maximum burst size of M_k packet arrivals in one slot. The buffers are assumed to be initially empty. A randomized stationary scheduler has zero-loss if and only if for each state $\mathbf{x}^i \in \Omega$, one of the following two conditions are satisfied.*

1. *The stationary distribution is zero for state $\mathbf{x}^i$,* i.e., $s_i = 0$.

2. *If z_k^i denotes the minimum number of packets transmitted from the k^{th} queue in state $\mathbf{x}^i$ using the given scheduler, then $(x_k^i - z_k^i) \le (L_k - M_k), \forall k$.*

Proof: See Appendix A.1. □

3. Optimal Schedulers

In the previous section, we defined the multiuser power control used and introduced the class of schedulers of interest. In this section we characterize the achievable region in the $(K+1)$ dimensional delay-power space in Theorems 2 and 3. We then compute optimal schedulers in the class Θ.

Definition 2 (Convex combination of schedulers) *When scheduler α equals the convex combination of 2 schedulers, say $\alpha \equiv \eta\alpha_A + (1-\eta)\alpha_B, 0 \le \eta \le 1$, it means that the number of packets $\mathbf{u}$ transmitted in each state $\mathbf{x}^i$ using scheduler α equals the number of packets transmitted in state $\mathbf{x}^i$ using scheduler α_A with probability η and the number of packets transmitted in the same state $\mathbf{x}^i$ using scheduler α_B with probability $1-\eta$.*

In Theorem 2, we prove that the average power and delays for an arbitrary zero loss randomized scheduler can be obtained as the convex combination of the power and delays of zero loss deterministic schedulers.

Theorem 2 (Characterization of delay and power) *Consider a set of K queues with finite buffer sizes L_k and K independent input processes with no more than M_k packet arrivals in one time-slot. For any randomized scheduler $\alpha \in \Theta$, there exists non-negative η_i with $\sum_{i=1}^{G} \eta_i = 1$, such that*

$$D_k(\alpha) = \sum_{i=1}^{G} \eta_i D_k(\gamma_i), \forall k = 1, \ldots, K \qquad (11)$$

$$P_{avg}(\alpha) = \sum_{i=1}^{G} \eta_i P_{avg}(\gamma_i) \qquad (12)$$

where γ_i is a zero loss deterministic scheduler and $G = |\Theta \cap \mathbb{W}|$. Also, $D_k(\alpha)$ represents the average delay of user k and $P_{avg}(\alpha)$ represents the total average power using scheduler α.

*P*roof: See Appendix A.2. □

In Theorem 3, we prove a converse to Theorem 2, that there exists a scheduler whose delays and power are the convex combination of the delays and powers of any two deterministic zero loss schedulers.

Theorem 3 (Converse to Theorem 2) *Let α_A and α_B be two zero loss deterministic schedulers with average delays $\mathbf{D}_{avg}(\alpha_A)$, $\mathbf{D}_{avg}(\alpha_B)$ and average powers $P_{avg}(\alpha_A), P_{avg}(\alpha_B)$ respectively. Then, there exists a zero loss scheduler which achieves average delay $\eta\mathbf{D}_{avg}(\alpha_A) + (1-\eta)\mathbf{D}_{avg}(\alpha_B)$ and average power $\eta P_{avg}(\alpha_A) + (1-\eta)P_{avg}(\alpha_B)$ for all $0 \le \eta \le 1$.*

Proof: The proof follows from time-sharing arguments and is given in Appendix A.3. □

It should be emphasized that the convex span of $\mathbb{W} \cap \Theta$ is larger than Θ (some schedulers in convex span can be non-zero loss). Given that the randomized scheduler performance can be obtained by appropriate linear combination of deterministic schedulers, the delay-power region can be characterized as follows.

Corollary 1 (Characterization of delay-power region) *The boundary of the achievable region in the $(K+1)$ dimensional delay-power space is piecewise planar with the vertices achieved by deterministic schedulers.*

Proof : The total number of deterministic zero loss schedulers is finite, and the delay-power performance of all randomized zero outage schedulers is given by a convex combination of delay-power of deterministic zero loss schedulers. Hence the corollary follows. □

To find the achievable delay-power region, we use a dynamic programming technique commonly known as Value Iteration Algorithm (VIA) [17]. The details of the VIA are given in Appendix A.4. The delay bound $\mathbf{D}_o$ is reduced monotonically (one element at a time, in any order) and the minimal power *deterministic* policy is found for each $\mathbf{D}_o$. Thus, instead of solving (4), we are required to solve the following optimization problem,

$$P^{\dagger}(\mathbf{D}_o) = \min_{\substack{\Theta \cap \mathbb{W} \\ \mathbb{E}[x_{k,n}] \leq \lambda_k D_{k,o} \forall k}} \lim_{n \to \infty} \mathbb{E}\{P_n\}. \quad (13)$$

The convex hull of the resulting $(\mathbf{D}_o, P^{\dagger}(\mathbf{D}_o))$ pairs gives the delay-power region, using Theorem 2 and Corollary 1. An example of an optimal scheduler obtained using VIA is given in Example 1.

Example 1 (Optimal Broadcast Scheduler) *Let* $K = 2,\ M_1 = 2,\ M_2 = 2,\ L_1 = 5,\ L_2 = 5$ *and* $N_1 = \frac{1}{9},\ N_2 = \frac{1}{7}$. *The scheduler is specified by the following two matrices representing the mappings* $\mathbf{x}_n \mapsto u_1$ *and* $\mathbf{x}_n \mapsto u_2$. *The number of packets transmitted from queue 1 (or 2) when* $x_{1,n} = i$ *and* $x_{2,n} = j$ *is given by the* $(i+1)^{st}$ *row and* $(j+1)^{st}$ *column of matrix* u_1 *(or* u_2*).*

$$u_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 \\ 3 & 3 & 3 & 3 & 2 & 1 \\ 3 & 3 & 3 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 2 \end{bmatrix} u_2 = \begin{bmatrix} 0 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

The resulting average delays are $D_1 = 1.0065$ *and* $D_2 = 1.8264$ *time-slots and average power required is 0.1226. Clearly, buffer 1 is almost always empty except when buffer 2 is full. In general we noticed that* u_1 *is non decreasing with increasing* x_1 *for a fixed* x_2. *We also found that for a fixed* x_2, *the increase in* u_1 *is no greater than one when* x_1 *increases by one,* i.e., $|u_1(x_1+1, x_2) - u_1(x_1, x_2)| \leq 1$.

The achievable delay-power region is given in Fig. 2 and explained in Sec. 4.2.

4. Low complexity scheduling

When there are large number of users in the system and their buffer sizes are large, the number of possible states in the VIA increases exponentially and finding the optimal scheduler becomes computationally intensive. Further, in situations where the optimal scheduler is adapted over time (say when arrival distributions are measured in real time) the implementation of the VIA could be difficult. In this section, we present low complexity near optimal schedulers. We also derive a closed form approximation for the dependence of power on delay of different users in Sec. 4.3.

4.1. Low complexity CDMA schedulers

One simple and obvious way to design a low complexity scheduler is to make decisions on the number of packets $u_{k,n}$ to transmit for each user independent of the other users and dependent only on their respective buffer states $x_{k,n}$. We refer to such a scheduler as a non-cooperative scheduler and use its performance as a benchmark to compare the performance of the proposed optimal and low complexity schedulers. As will be clear from the numerical examples in Sec. 4.2, there could be a significant difference in the performance of the optimal and non-cooperative schedulers.

We now propose a set of low complexity schedulers which reduces the number of states in the VIA and consists of two steps as detailed below.

1. In the optimal scheduler, $u_{k,n}$ is a function of $\mathbf{x}_n$. In the proposed low complexity scheduler, we force $u_{k,n}$ to be a function of $x_{k,n}$ and $f_k(\mathbf{x}_n)$ where $f_k(.)$ is a discrete function of $\mathbf{x}_n$ and does not depend on $x_{k,n}$. For example, when $K = 2$, we choose

$$f_1(\mathbf{x}_n) = \begin{cases} 1 & \text{if } x_{2,n} \leq \lambda_2 D_{2,o} \\ 2 & \text{if } x_{2,n} > \lambda_2 D_{2,o} \end{cases},$$

   and similarly for $f_2(\mathbf{x}_n)$. In this case, we have chosen $f_1(\mathbf{x}_n)$ to take on two different values. As the number of values taken by $f_1(\mathbf{x}_n)$ increases to be equal to (L_2+1) and $f_1(\mathbf{x}_n) = x_{2,n}$ the low complexity scheduler approaches the optimal scheduler. At the other extreme, as the number of values taken by $f_1(\mathbf{x}_n)$ equals 1, the low complexity scheduler chooses $u_{k,n}$ only as a function of $x_{k,n}$ and essentially becomes a non-cooperative scheduler. Hence, this method defines a whole range of schedulers in which the performance can be traded with the computational complexity.

   We can modify the VIA to compute *pseudo-optimal* solutions with the constraints as imposed in condition 1. However, to further reduce the complexity of computing the schedulers, we propose the following step.

2. For each user, choose $u_{k,n}(x_{k,n}, f_k(\mathbf{x}_n)) = \lfloor \log(\beta_{k,i} x_{k,n}) \rfloor$ where $f_k(\mathbf{x}_n) = i$ and $\lfloor . \rfloor$ represents the largest integer no greater than the argument. We further impose the constraints that $0 \leq u_{k,n} \leq x_{k,n}$ and $u_{k,n} \geq M_k - L_k + x_{k,n}$ (needed to maintain zero buffer loss). For example, with $f_1(\mathbf{x}_n)$ defined as in condition 1 above, we let $u_{1,n}(x_{k,n}, f_1(x_n)) = \lfloor \log(\beta_{k,1} x_{k,n}), \lfloor \log(\beta_{k,2} x_{k,n}) \rfloor$ when $f_1(\mathbf{x}_n) = 1$ and 2 respectively. The motivation for this step 2, comes from the near optimality of the *log-linear* scheduler in the single user scenario [14, 16].

We now prove that these low complexity schedulers can achieve the whole range of delays for the different users.

Proposition 1 (Achievability of entire delay range) *There exist parameters $\{\beta_{k,i}\}_{k,i}$ such that the whole range of delays are achievable for all users.*

Proof: By imposing the constraint that $\beta_{k,i} = \beta_k \forall i$, we can easily see that the resulting delay for user k depends only on β_k and not the other $\beta_j, j \neq k$. For very small value of β_k such that $\lfloor \log(\beta_{k,1} L_k) \rfloor < 1$ the resulting scheduler will have largest possible average buffer length while still maintaining zero buffer overflow (because we ensure that $u_{k,n} \geq M_k - L_k + x_{k,n}$.). Also for very large value of β_k such that $\lfloor \log(\beta_{k,1} M_k) \rfloor = M_k$, a delay of exactly one can be achieved. Now, by monotonically increasing β_k in the range suggested above, the whole range of possible delays for user k can be achieved [16]. □

We now present a method for choosing the $\beta_{k,i}$ parameters given a particular set of delay constraints in the special case of 2 users and $f_k(.)$ as illustrated in step 2 above.

The structure of the optimal schedulers computed using the VIA indicates that in general for a given buffer state of user 1, we transmit more (or equal) packets when buffer state of user 2 is sparsely filled than when buffer of user 2 is almost full, *i.e.*, $u_1(x_1, i) \geq u_1(x_1, j)$ if $i < j$: This scheduling behavior is also intuitive since the total power depends exponentially on both rates. Hence, we further impose the constraint that $\beta_{1,1} \geq \beta_{1,2}$ in the low complexity scheduler in the two user case.

Now, $\beta_{k,1}$ is varied by a predetermined fixed steps starting from $\beta_{k:min}$ to $\beta_{k:max}$. For each value of $\beta_{k,1}$, the value of $\beta_{k,2}$ is varied using the same fixed step size between its minimum and maximum value with the further constraint that $\beta_{k,1} \geq \beta_{k,2}$. For each value of $\beta_{k,1}$ and $\beta_{k,2}$ the required transmit power and the delays achieved are computed using (9) and (8) respectively. Now from the total available delays and powers, find the largest delay achieved that is no greater than the delay bound specified; the corresponding power is the required transmission power using the low complexity scheduler. The values for $\beta_{k:min}$ and $\beta_{k:max}$ are determined based on simple discrete properties of $\lfloor \log(.) \rfloor$ functional and ensuring that $u_{k,n} \leq x_{k,n}$. For the simulation results shown in Sec. 4.2, we chose $\beta_{k:min} = \frac{e}{L_k - M_k}$ and $\beta_{k:max} = \frac{1}{L_k} e^{L_k}$. The delay-power profile achieved using this low complexity scheduler is shown in Fig. 2 and is explained in the next section.

4.2. Numerical Results

A cross-section of the achievable delay-power region is given in Fig. 2 for a two-flow ($K = 2$) system. The delay of user 2 (QoS flow) is fixed at 1 time-slot and the variation of the total power with the delay of user 1 (elastic flow) is plotted. Again, it should be reiterated that the optimal scheduler achieves the delays and powers on the boundary of this achievable region.

For unity delay bound, $D_{k,o} = 1$ for both flows, the scheduling action is straightforward; all packet arrivals during a slot must be transmitted during the next slot. As the delay of user 1 increases, the total transmit power reduces and the reduction in power is significant for the initial increases in delays. The reduction in power is due to two components: i) Convex relation between power and rate of transmission for each user (9). Transmitting more packets requires exponentially increasing amounts of power. As the delay deadline of user 1 increases, its packets are delayed so that transmission of large values of u_1, is minimized and thus average power consumption is reduced. In this case, user 2's packets have to be instantaneously transmitted since it has unit delay bound. ii) Statistical multiplexing of the packets of different users. Since the power depends exponentially on the sum rate of all users (9), when user 2 has a large number of packets to transmit, user 1 throttles back its transmissions and sends small u_1. Similarly, when u_2 is small, user 1 increases its transmission rate (since its packets are buffered) to meet its delay bounds. Clearly, the statistical multiplexing gain is larger for larger delays. As the delays of all users increases, the optimal scheduler, transmits close to the average arrival rate (subject to availability of packets and meeting of delay bounds) for each user in each time-slot and the required power equals the Shannon limit $P_{bc}(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$=$[\lambda_1, \ldots, \lambda_K]$ is the vector of average arrival rates. In Fig. 2, even as delay of user 1 increases, the average power does not reach P_{bc}; it is achieved only when the delay of both users asymptotically go to infinity.

The performance of the noncooperative and low complexity schedulers are also shown in Fig. 2. The non-cooperative scheduler requires appreciably more power than the optimal scheduler for all $D_{2,o} > 1.1$. In fact, as the delay of the elastic flow increases, the non-cooperative scheduler requires approximately 30% more power than the optimal scheduler ($L_1 = L_2 = 20, M_1 = M_2 = 6, N_1 = 0.01, N_2 = 0.25$). The reasons for such large power disparity are as follows. For larger delays, the non-cooperative scheduler will transmit close to mean arrival rate for the elastic flow, irrespective of how many packets are transmitted for the QoS flow. In contrast, for the optimal scheduler, whenever the QoS flow is forced to transmit a large number of packets to meet its delay bound, the scheduler only transmits a small number of packets for the elastic flow. Alternately, when the QoS flow has few packets to send, the elastic flow can be opportunistic and send more packets. Thus, as noted earlier, the optimal scheduler statistically multiplexes the two flow-data.

A sample realization, in Fig. 3, of the number of packets transmitted by the different schedulers for both flows confirms the above observation. The total number of pack-

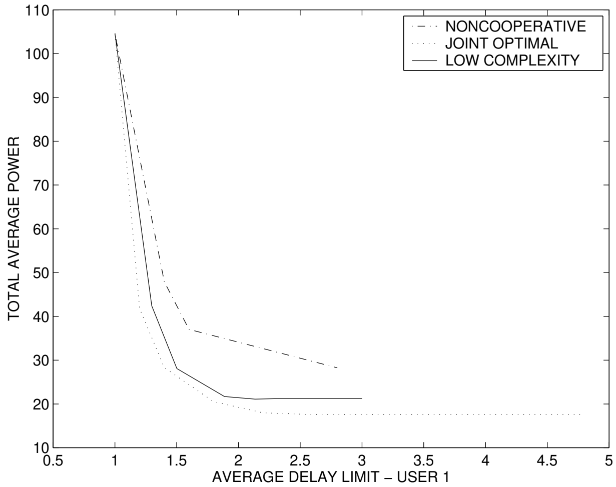

Figure 2. Performance of optimal, low complexity and noncooperative schedulers for two users. The delay for user 2 is fixed at 1 time-slot. ($L_1 = L_2 = 20$, $M_1 = M_2 = 6$).

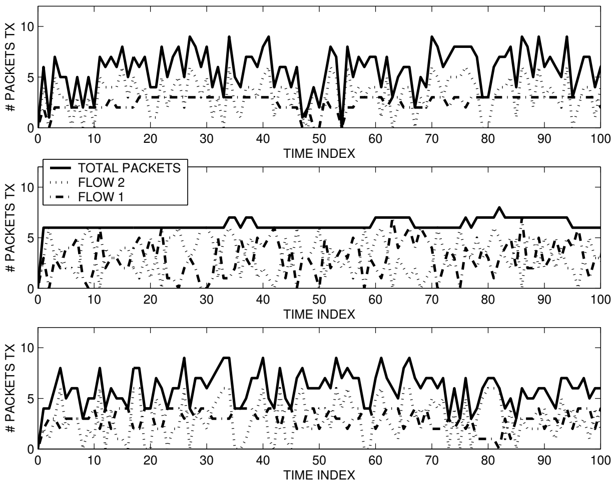

Figure 3. Sample run for non-cooperative, optimal and low complexity CDMA schedulers for two-flows with different delay requirements. ($L_1 = L_2 = 20$, $M_1 = M_2 = 6$). The variance of $u_1 + u_2$ in this sample run for the joint optimal, low complexity and non-cooperative schedulers are 0.44, 2.6 and 4.2 respectively.

ets transmitted by the optimal scheduler is almost a constant, thereby avoiding large power swings. However, the non-cooperative scheduler transmits approximately constant number of packets in each time-slot for the elastic flow, leading to large variations in the total number of transmitted packets and hence the total required power. The optimal scheduler decisions for the elastic flow are strongly tied to the the decisions made for the QoS flow, which is not the case in the non-cooperative scheduler. The low complexity CDMA scheduler achieves a balance between approaching the performance of optimal scheduler and increasing the complexity of computing the optimal scheduler. By choosing $\beta_{k,1} > \beta_{k,2}$ the low complexity scheduler transmits more packets for user k for the same buffer state of user k, when the other users buffer is less full and vice versa. Although the total number of packets transmitted using low complexity scheduler (see Fig. 3) is not a constant (like in optimal scheduler), it has much lesser variation than in non-cooperative scheduler. The sample variance of $(u_1 + u_2)$ in the optimal, low complexity and non-cooperative schedulers are 0.44, 2.6 and 4.2, respectively, confirming our explanation. The power-delay performance of the low complexity scheduler is near optimal as is evident from Fig. 2.

4.3. Closed form approximation

We now derive an approximate relationship between the total transmission power, the delays and rates of each user in a broadcast system. One of the motivating factors in deriving such a relationship is to allow coarse approximation of the power required to support a set of users with a given set of delay constraints. This approximate power could be used to make admission control decisions based on the total available power. Alternatively, one could calculate the excess delay needed to support the users under given average power constraints. The relationship (derivation in Appendix A.5) is given by

$$P_o \approx W \sum_{i=1}^{K} \left(e^{\frac{\lambda_i + \frac{\sigma^2_{a_i}}{4\lambda_i D_i - 2}}{W}} - 1 \right) N_i \prod_{j=i+1}^{K} e^{\frac{\lambda_j + \frac{\sigma^2_{a_j}}{(4\lambda_j D_j - 2)}}{W}}, \tag{14}$$

where $\sigma^2_{a_i}$ represents the variance of the arrival process $\{a_i\}$ of user i. The effect of random source arrivals is clearly highlighted in (14) by the factor $\frac{\sigma^2_{a_i}}{\lambda_i}$. Sources which exhibit large variations in their arrivals compared to their mean rate and are "more" bursty, have large $\frac{\sigma^2_{a_i}}{\lambda_i}$. For more bursty sources, even small increase in D_{avg} allows considerable power savings; practical examples of such sources include web and email traffic. Asymptotically, as $D_{avg} \rightarrow \infty$, the high delay approximation gives (6) with transmission rates λ, which is the same as the Shannon limit. As $D_{avg} \rightarrow \infty$, the random arrivals of the source are completely smoothened and hence well known information theoretic analysis applies. The numerical accuracy of the approximation is studied in [14].

5. Conclusions

In this paper, we presented optimal power efficient packet schedulers for broadcast channels that meet certain average delay bounds. We also introduced an approximate relation between power, rates and delays for multiuser broadcast Gaussian channel that is accurate for high delays; To the best of our knowledge its the first such closed form relationship.

This paper assumes a simple memoryless model for source arrivals. Extensions to more complicated traffic models and situations where source arrival statistics are measured need to be addressed in future work. Also, the schedulers presented in this paper could form the basis for efficient scheduling in multihop networks.

Acknowledgments

This work was supported in part by Nokia Corporation and by the NSF award 0311398.

References

[1] M. Agarwal and A. Puri. Base station scheduling of requests with fixed deadlines. In *Proc. of INFOCOM*, 2002.

[2] I. F. Akyildiz, D. A. Levine, and I. Joe. A slotted CDMA protocol with BER scheduling for wireless multimedia networks. *IEEE/ACM Trans. Networking*, 7(2):146–158, 1999.

[3] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. CDMA data QoS scheduling on the forward link with variable channel conditions. *Technical Report, Bell Laboratories, Lucent Technologies.*, 2000.

[4] R. A. Berry and R. G. Gallager. Communication over fading channels with delay constraints. *IEEE Trans. on Information Theory*, 48(5):1135–1149, May 2002.

[5] D. P. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[6] V. Bharghavan, S. Lu, and T. Nandagopal. Fair scheduling in wireless packet networks: Issues and approaches. *IEEE Personal Comm. Mag.*, 6(1):44–55, Feb. 1999.

[7] B. Collins and R. L. Cruz. Transmission policies for time varying channels with average delay constraints. In *Proc. Allerton Intnl. Conf. on Comm., Control and Computing*, pages 709–717, Monticello, IL, 1999.

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[9] S. Hanly and D. Tse. Multi-access fading channels: Part II: Delay-limited capacities. *IEEE Trans. on Information Theory*, 44(7):2816–2831, November 1998.

[10] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram. Downlink scheduling in CDMA data networks. In *Proceedings of MOBICOM*, pages 179–190, 2000.

[11] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Downlink power allocation for multi-class cdma wireless networks. In *Proc. of INFOCOM*, 2002.

[12] P. Marbach and R. Berry. Downlink resource allocation and pricing for wireless networks. In *Proc. of INFOCOM*, 2002.

[13] B. Prabhakar, E. U. Biyikoglu, and A. E. Gamal. Energy-efficient transmission over a wireless link via lazy packet scheduling. In *Proc. of INFOCOM*, Anchorage, Alaska, April 2001.

[14] D. Rajan. *Power efficient transmission policies for multimedia traffic over wireless channels*. PhD thesis, Rice University, April 2002.

[15] D. Rajan, A. Sabharwal, and B. Aazhang. Delay and rate constrained transmission policies over wireless channels. In *Proc. GLOBECOM*, San Antonio, TX, November 2001.

[16] D. Rajan, A. Sabharwal, and B. Aazhang. Delay bounded packet scheduling of bursty traffic over wireless channels. *IEEE Trans. on Information Theory*, 50(1):125–144, 2004.

[17] L. I. Sennott. *Stochastic Dynamic Programming and the control of Queueing Systems*. John Wiley and Sons, Inc., 1999.

A Appendix

A.1. Proof of Theorem 1: Zero Buffer Overflow

If $x^i_{k,n} \leq L_k - M_k$ then from (2), $x^j_{k,n+1} \leq L_k$ and as a result, there will be no buffer overflow. So a buffer overflow is possible only if $x^i_{k,n} > L_k - M_k$, thus we will focus on the case when $x^i_{k,n} = L_k - M_k + 1, ..., L_k$.

Direct (if) : We will first prove that if either of Condition 1 or 2 is satisfied, then the scheduler will be zero loss. If $x^i_{k,n} \geq L_k - M_k$, and Condition 2 is satisfied, then using (2) we get

$$\begin{aligned} x^j_{k,n+1} &\leq x^i_{k,n} - z_k + M_k \\ &\leq L_k - M_k + M_k = L_k. \end{aligned}$$

Therefore there will be no buffer overflow. On the other hand, if Condition 1 is satisfied for state $x^i_{k,n} \geq L_k - M_k$, then there is zero probability for the buffer to be in that state. Thus, no buffer overflow is possible. Note that we have assumed that the buffer starts from the zero state, $x^i_{k,0} = 0$ (if the buffer was initialized from a state $x^i_{k,n} = y$ such that $s_i = 0$ and $(y - z_k) > L_k - M_k + 1$ then there will be a buffer overflow with finite probability).

Converse (only if) : We prove by contradiction that if there exists state $x^i_{k,n} = y \geq L_k - M_k$ that does not satisfy both Conditions 1 and 2, then the corresponding scheduler will not be zero loss. If state i of the buffer does not satisfy Condition 1, with finite non-zero probability the buffer will be in state i. Since state i also does not satisfy Condition 2, using (2) with finite probability

$$\begin{aligned} x^j_{k,n+1} &= y - z_k + M_k \\ &> L_k - M_k + M_k = L_k. \end{aligned}$$

Consequently, a buffer overflow occurs. □

A.2. Proof of Theorem 2: Characterization of delay and power

The proof consists of 3 main steps as follows.

1. We first provide a constructive method of expressing a randomized zero loss schedulers α in terms of the deterministic schedulers.

   (a) Set user number $k = 1$.

   (b) Set state $i = 1$; If $k = 1$, let the *parent* scheduler (denoted β) be the original scheduler.

   (c) For each state $\mathbf{x}^i$, the *parent* scheduler of interest can be written as convex combination of x^i_k different *children* schedulers (denoted ρ_j) which are identical for all states for all users except user k and for user k identical in all states except state $\mathbf{x}^i$, *i.e.*, $\beta \equiv \sum_{j=1}^{x^i_k} \tau_j \rho_j$, where $\sum \tau_j = 1, \tau_j \geq 0$. In state $\mathbf{x}^i$, the output for user k in children scheduler ρ_j is j. The factors τ_j in the convex combination equal the probability of transmitting $0, 1, \ldots, |x^i_k|$ packets for user k in state $\mathbf{x}^i$ using parent scheduler β, *i.e.*, $\tau_j = p(u_k = j|\mathbf{x}^i)$.

   (d) Let $i = i + 1$; Repeat Step c for each of the children scheduler in the previous step if $i < |\Omega|$.

   (e) Let $k = k + 1$. Repeat Steps b-d for each of the children scheduler in the previous step if $k < K$.

   Further in the above expansion of the scheduler α, the coefficients of all non-zero loss deterministic schedulers can be shown to be zero. (Proof of this follows directly from Proposition 6 of [16].)

2. We now prove the following Lemma.

   Lemma 1 (D-P of 2 "almost" identical schedulers) *Let α_A and α_B be two zero outage schedulers which have identical output for all users in all buffer states except in the $\mathbf{x}^i$ buffer state. Moreover, in the i^{th} buffer state let schedulers α_A and α_B differ in the output of only one user (we refer to such schedulers as "almost" identical schedulers). The delay and power of scheduler α which is given by a convex combination of α_A and α_B, equals a convex combination of the delays and powers of α_A and α_B, respectively. In other words, if $\alpha \equiv \eta\alpha_A + (1-\eta)\alpha_B$, $\eta \in [0,1]$ then $D_k(\alpha) = \eta D_k(\alpha_A) + (1-\eta')D_k(\alpha_B)$ and $P_{avg}(\alpha) = \eta' P_{avg}(\alpha_A) + (1-\eta')P_{avg}(\alpha_B)$, where $0 \leq \eta' \leq 1$.*

   Proof: It is easy to see that $\mathbb{C}_{\alpha_A}$ and $\mathbb{C}_{\alpha_B}$ differ in only the i^{th} column where $\mathbf{x}^i$ is the state in which scheduler actions α_A and α_B differ. Moreover, $\mathbb{C}^i_\alpha$ which represents the i^{th} column of $\mathbb{C}_\alpha$ is given by $\mathbb{C}^i_\alpha = \eta\mathbb{C}^i_{\alpha_A} + (1-\eta)\mathbb{C}^i_{\alpha_B}$. Using simple algebraic manipulation, the stationary probability $\mathbf{s}_\alpha$ using scheduler α (given by (7)) can be shown to be $\mathbf{s}_\alpha = \eta'\mathbf{s}_{\alpha_A} + (1-\eta')\mathbf{s}_{\alpha_B}$ where

   $$\eta' = \frac{\eta \mathbf{s}_{\alpha_A,i}}{\eta \mathbf{s}_{\alpha_A,i} + (1-\eta)\mathbf{s}_{\alpha_B,i}}. \tag{15}$$

   It follows from (8) that the average delay of user k using scheduler α, is given by $D_k(\alpha) = \eta D_k(\alpha_A) + (1-\eta')D_k(\alpha_B)$. Clearly, the delays of the other users are the same in schedulers α_A, α_B and consequently in α.

   Now it remains to show that the average power of scheduler α follows a similar convex combination. From (9) we find that

   $$\begin{aligned} P_{avg}(\alpha) &= \sum_{\mathbf{x^i}}\sum_{\mathbf{u}} \mathbf{s}_{\alpha,i} P_{bc}(\mathbf{u}) p_\alpha(\mathbf{u}|\mathbf{x}^i) \\ &= \eta' P_{avg}(\alpha_A) + (1-\eta')P_{avg}(\alpha_B) \end{aligned}$$

   which follows after simple manipulation and application of (15). Hence, the lemma is proved.

3. In our construction of the basis expansion of scheduler α in the first step, we have a convex combination of schedulers which differ only in the output of a single user k for a particular buffer state $\mathbf{x}^i$. Hence, using Lemma 1 the delay and power of scheduler α is given by a convex combination of the delays and powers of two schedulers (which need not be deterministic). Each of these schedulers is further written in terms of schedulers which are "almost" identical and hence its power and delay are convex combinations of delay and power of some schedulers. Repeating this process till the resulting schedulers are deterministic, the theorem is proved. □

A.3. Proof of Theorem 3: Converse of Theorem 2

Since scheduler α_A achieves average delay $\mathbf{D}_{avg}(\alpha_A)$ and average power $P_{avg}(\alpha_A)$, it implies that for any $\epsilon > 0$, there exists an $N > N_{\alpha_A}$ such that using scheduler α_A for N time-slots gives average delay $\hat{\mathbf{D}}_{\alpha_A}$ and average power $\hat{P}_{\alpha_A}$ satisfying $|\hat{\mathbf{D}}_{avg}(\alpha_A) - \mathbf{D}_{avg}(\alpha_A)| < \epsilon$ and $|\hat{P}_{avg}(\alpha_A) - P_{avg}(\alpha_A)| < \epsilon$. Similarly N_{α_B} exists for α_B.

Now, we time share between schedulers α_A and α_B in the ratio $\eta : (1-\eta)$ to get scheduler α. Given any $\epsilon > 0$ find $N = \max\left(\frac{N_{\alpha_A}}{\eta}, \frac{N_{\alpha_B}}{1-\eta}\right)$. While switching from scheduler α_A to α_B, we additionally flush the buffer in time-slot ηN. Now,

$$\begin{aligned} |\hat{\mathbf{D}}_{avg}(\alpha) - \mathbf{D}_{avg}(\alpha)| &\leq \eta|\hat{\mathbf{D}}_{avg}(\alpha_A) - \mathbf{D}_{avg}(\alpha_A)| + \\ (1-\eta)|\hat{\mathbf{D}}_{avg}(\alpha_B) &- \mathbf{D}_{avg}(\alpha_B)| + \frac{L_{max}}{N\lambda_{min}} \\ &\leq \eta\epsilon + (1-\eta)\epsilon + \frac{L_{max}}{N\lambda_{min}} \\ &= \epsilon + \frac{L_{max}}{N\lambda_{min}}, \end{aligned}$$

where $L_{max} = \max(L_1, L_2, \ldots, L_K)$ and $\lambda_{min} = \min(\lambda_1, \ldots, \lambda_K)$. In the limit as $N \to \infty$, $\frac{L_{max}}{N\lambda_{min}} \to 0$. Hence, $\hat{\mathbf{D}}_{avg}(\alpha) \to \eta\mathbf{D}_{avg}(\alpha_A) + (1-\eta)\mathbf{D}_{avg}(\alpha_B)$. Similarly,

$$\begin{aligned} |\hat{P}_{avg}(\alpha) - P_{avg}(\alpha)| &\leq \eta|\hat{P}_{avg}(\alpha_A) - P_{avg}(\alpha_A)| + \\ (1-\eta)|\hat{P}_{avg}(\alpha_B) &- P_{avg}(\alpha_B)| + \frac{1}{N}P_{bc}(\mathbf{L}) \\ &\leq \eta\epsilon + (1-\eta)\epsilon + \frac{1}{N}P_{bc}(\mathbf{L}) \\ &= \epsilon + \frac{1}{N}P_{bc}(\mathbf{L}), \end{aligned}$$

where $\mathbf{L} = [L_1, L_2, \ldots, L_K]$. In the limit as $N \to \infty$, $\frac{1}{N}P_{bc}(\mathbf{L}) \to 0$. Again, if we consider K consecutive blocks of N time-slots, the total average delay and power is given by the average of the delays and powers in each block of N time-slots. Hence, by time-sharing between schedulers α_A and α_B we can achieve all delays and powers given by the convex combination of the delays and powers of α_A and α_B. $\square$

A.4. Value Iteration Algorithm (VIA)

Let $C(i, a)$ represent the cost incurred in doing action a in state i. Let $P_{ij}(a)$ is the probability of transitioning from state i to state j under action a. Also, let x denote a predetermined state and ϵ is a small positive number that determines the stopping criterion. The general steps in the Value Iteration Algorithm [17] are

1. Initialize $v_0 \equiv 0$, $\delta = 1$ and $k = 0$.
2. Evaluate $w_k(i) = \min_a \left\{C(i, a) + \sum_j P_{ij}(a)v_k(j)\right\}$.
3. Set $\delta = |w_k(x) - w_{k-1}(x)|$ and $v_{k+1}(i) = w_k(i) - w_k(x)$.
4. Repeat steps 2 and 3 above till $\delta < \epsilon$.
5. The optimal actions for each state i are obtained as

$$a^*(i) = \arg\min_a \left\{C(i, a) + \sum_j P_{ij}(a)v_k(j)\right\}.$$

In the above, k represents the k^{th} iteration of the VIA. In the case of schedulers which bound average delay, the different VIA symbols have the following correspondence, $i \equiv x^i$, $a \equiv u(x^i)$. Also, $P_{i,j}(a)$ depends on the arrival distribution $p(a_n)$ and $C(i, a) = P_{bc}(\mathbf{u}_n) + \varepsilon x_n$ where ε is the Lagrangian used in (13).

A.5. Multiuser power-delay approximation

The derivation of the multiuser power-delay approximation is based on two simplifying assumptions: the packets are infinitely divisible (fluid model) and the output distribution of number of packets transmitted in a time-slot for each user is Gaussian and independent of other users. The fluid model allows us to divide packets arbitrarily, and thus avoid the discrete nature of scheduler decisions, which is a significant reason for intractability of closed-form analysis of the optimal scheduler.

As discussed in Sec. 4.2, for large $D_{k,o}$, the optimal scheduler transmits at rates close to average arrival rate λ_k if buffer overflows can be avoided. The scheduler action $u_{k,n}(\mathbf{x}_n)$ for user k can thus be approximated around $\mathbb{E}\{\mathbf{x}_n\}$ by the following linear relation,

$$u_{k,n} = \mu_k x_{k,n} + \nu_k. \tag{16}$$

For a stable system, the average arrival rate should equal the average departure rate and hence $\mathbb{E}\{u_{k,n}\} = \lambda_k$. Using (2) and the stationarity of $x_{k,n}$ we find $\mu_k \mathbb{E}[x_{k,n}] = -\nu_k + \lambda_k$. The slope of the scheduler, ν_k, at $\mathbb{E}\{x_{k,n}\}$ is given [16] as $\mu_k = 1/\mathbb{E}\{x_{k,n}\}$. The variance of the output process, $u_{k,n}$, denoted by σ_{u_k}, is found using (16) as,

$$\sigma^2_{u_k} = \frac{\sigma^2_{a_k}\mu_k}{2 - \mu_k} = \frac{\sigma^2_{a_k}}{2\mathbb{E}[x_{k,n}] - 1}, \tag{17}$$

where $\sigma^2_{a_k}$ is the variance of the arrival process $\{a_{k,n}\}$. Using the Gaussian approximation for the distribution of $u_{k,n} \sim \mathcal{N}(\lambda_k, \sigma^2_{u_k})$, the average transmit power is

$$\mathbb{E}\{P_n\} = \left(\prod_{k=1}^{K} \frac{1}{\sqrt{2\pi\sigma^2_{u_k}}}\right) \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} W \sum_{i=1}^{K}$$

$$\left(e^{\frac{u_{i,n}}{W}} - 1\right) N_i \prod_{j=i+1}^{K} e^{\frac{u_{j,n}}{W}} \prod_{l=1}^{K} e^{-\frac{(u_{l,n}-\lambda_l)^2}{2\sigma^2_{u_{l,n}}}} du_{1,n} \ldots du_{K,n}$$

which can be evaluated and shown to equal (14).