Adaptive Transmission of Variable-Rate Data over a Fading Channel for Energy-Efficiency

Elif Uysal-Biyikoglu, Abbas El Gamal and Balaji Prabhakar
Information Systems Laboratory
Stanford University
Stanford, CA 94305
elif@stanford.edu, abbas@isl.stanford.edu, balaji@stanford.edu

Abstract— **This paper explores the adaptation of transmission rate and power jointly to the data generation rate and channel fading, for minimizing transmission energy. The optimal offline adaptation problem is solved, which provides a lower-bound on the transmission energy consumed by any practical, that is, online, scheme. A heuristic online algorithm, Look-ahead Water-filling, is developed for adapting to the queue state as well as the channel state, and is shown through simulations to achieve transmission energy per packet close to optimal. As the packet arrival rate is varied within known limits, the average energy per packet used by Look-ahead Water-filling is significantly lower than that achieved by optimal adaptation to the channel only (water-filling in time). The delay per packet is larger, but is almost constant for all data arrival rates. The results can be generalized to multi-access and broadcast fading channels.**

I. Introduction

Adapting to a time-varying channel to maximize the average information rate for a given power constraint is a well-understood problem. Important studies (*e.g.*, [1], [2], [3]) have developed optimal rate and power allocation schemes for the single user and multiple-access fading channels, which can be approximated by practical adaptive coding/modulation techniques ([4], [5]). These previous studies were based on the assumption that data is always available or is generated continuously at a known rate. However, in many wireless data applications, the rate at which data is generated and needs to be transmitted is variable in time (*e.g.*, wireless web sessions or a sensor network where data gets generated at random times at each node). Schemes that ignore this variability and adapt solely to the channel can be inefficient in their usage of transmit power and bandwidth.

To understand how inefficiency may arise, consider the following data communication situation: the transmitter and receiver engage in a video conference or a web session, or they may alternate between the two. Different types of sessions have different rates of generating data packets, which are collected in the transmitter's buffer to be sent to the receiver. Let the rate at which packets arrive into the transmitter's buffer at time t be $\lambda(t)$ packets/second. These packets are transmitted to the receiver at a rate $\mu(t)$ packets/second. Now, assume we set $\mu(t) = \mu$, a constant that is large enough, say, for a high rate streaming video session. When the required rate drops, for example because the user switches to a lower-rate web session where $\lambda(t) \ll \mu$, the transmitter will idle a significant fraction of time and transmit unnecessarily fast the rest of the time, which is wasteful in terms of energy.

Schemes that adapt solely to the channel state can maximize the throughput for a given energy constraint. However, since they cannot track the value of $\lambda(t)$, they do not have control over delay (see, for ex., [8]). In order to guarantee finite average delay, they need to be set for the largest possible value of $\lambda(t)$, which causes them to be energy-inefficient. In this paper, our goal is to exhibit schemes that adapt $\mu(t)$ to $\lambda(t)$. We argue that this is essential to have optimal performance with respect to the three important metrics: energy, throughput, and delay.

II. The Problem Setup

As in [6], we have the following model of a single transmitter-receiver pair: Packets arrive at the transmitter's buffer at random times t_i, such that $t_1 = 0$, and $t_{i+1} \geq t_i$, $i \geq 1$. Packets are of length B bits. Each packet needs to be transmitted to the receiver, possibly using a different code rate r_i bits/symbol, corresponding to a transmission duration $\tau_i = B/r_i$ channel uses (symbols). The sequence $\{\tau_i\}$ will be called a *schedule*.

The channel is a discrete-time AWGN channel such that the received symbol at any time is $y = \sqrt{s}x + n$ where x is the transmitted symbol, $\sqrt{s}$ is the channel gain and n is Gaussian noise with power N. In order to communicate reliably at rate r over this channel, the capacity must be larger than r, hence the average transmit signal power must be at least $\frac{N}{s}(2^{2r} - 1)$. It will be convenient to call this power $f(r)/s$ and note that it is convex and monotonically increasing in r.

We make the block-fading assumption where the power gain s changes every T_c time units (the "coherence window") and assume that the power gains of different coherence windows are independent and identically distributed. For simplicity, we will also assume that the value of s is known to the transmitter and the receiver at the beginning

0-7803-7632-3/02/$17.00 ©2002 IEEE

of each coherence window.

The packet input process into the transmitter buffer at time t has the instantaneous rate $\lambda(t)$ packets/unit time. The time average arrival rate $\lambda \triangleq \lim_{T\to\infty} \frac{1}{T}\int_0^T \lambda(t)dt$ is bounded such that $\lambda < \lambda_{\max}$ with probability 1. We are interested in schedules that are stable, *i.e.*, scheduling algorithms that ensure that the number of packets in the buffer is finite with probability 1. Under this condition, we would like to minimize the transmission energy per packet. We will first exhibit the minimum-energy finite-horizon *offline* schedule, *i.e.*, when all the packet arrival times and channel states in a time window of finite length are known ahead of time. We will then consider online (*i.e.*, causal) scheduling algorithms and explore how much they approach the energy-efficiency of the optimal offline schedule. The first of these will be an algorithm that adapts optimally to the channel state by the well-known "water-filling in time" scheme [1]. The second online schedule we consider is what we call "Look-ahead Water-filling", which is a heuristic for jointly adapting to the arrival rate and the channel state.

III. Optimal Offline Scheduling

Consider the first m packet arrivals into the buffer, starting at time $t_1 = 0$. Among all scheduling algorithms that transmit these packets by time $T > 0$, the optimal offline schedule is the one that minimizes the total packet transmission energy, given perfect knowledge of the packet arrival instants and channel state values for the entire duration $[0, T)$ at time 0.

Define an "epoch" to be a time interval that starts with either a packet arrival or a change in the channel state, and continues until the next arrival or state change. The first epoch starts at $t_1 = 0$, and continues until t_2 or T_c, whichever is smaller, at which point the second epoch starts, and so forth. Let b_j denote the number of bits that have arrived at the beginning of epoch j, so $b_j = B$ if the j^{th} epoch starts with a packet arrival, and $b_j = 0$ otherwise. Let the duration of epoch j be ξ_j.

Lemma 1: In an optimal offline schedule, rate is constant during any epoch.

Proof: Suppose rate is r_1 in the first τ_1 time units of an epoch of length t, and r_2 during the remaining $t - \tau_1$. The transmit energy in this epoch is then $\tau_1 f(r_1)/s + (t - \tau_1)f(r_2)/s$ where s is the fading state during the epoch; note that by definition, fading state is constant during each epoch. The same number of bits can also be transmitted using the uniform rate $(r_1\tau_1 + r_2(t-\tau_1))/t$ for the whole time t (note that by definition all those bits are available at the beginning of the epoch). This new rate results in a total energy $tf((r_1\tau_1 + r_2(t-\tau_1))/t)/s$, which, by convexity of f, is strictly lower than previous, unless $r_1 = r_2$. ■

From Lemma 1, $\{r_j\}_{j=1}^{n}$ (n is the number of epochs) is sufficient to characterize the optimal schedule. With that, we are ready to express the offline scheduling problem:

$$\text{Minimize:} \quad \sum_{j=1}^{n} \xi_j f(r_j)/s_j,$$

$$\text{subject to:} \quad \sum_{j=1}^{k} r_j\xi_j \le \sum_{j=1}^{k} b_j \quad k = 1, \ldots, n,$$

This convex optimization problem can be solved efficiently by the iterative algorithm described below. This is a slightly modified form of the *FlowRight* algorithm of [7], hence we refer to it by the same name.

FlowRight Algorithm:

This is an iterative algorithm. In the beginning, the rates are set to $r_j^0 = b_j/\xi_i$, $i = 1, 2, \ldots, n$. Now, consider the first two epochs. The total number of bits transmitted in these two data epochs is $r_1^0\xi_1 + r_2^0\xi_2$. Keeping this total number of bits fixed, update r_1^0 to r_1^1, the value that minimizes the total energy of the first pair of data epochs. Note that $r_1^1 \le r_1^0$, since from their initial condition bits can only be pushed to the right (otherwise causality would be violated.) We therefore have to reset r_2^0 to a new value which is larger than (or equal to) its initial value.

Moving to the second pair of epochs, this time optimally decrease r_2^0 to r_2^1, and reset the value of r_3^0. Proceed in this way to obtain r_i^1 for $i = 1, \ldots, n$. This completes the first *pass* of the algorithm. After the first pass is complete, start from the beginning and update the rates of two adjacent data epochs at a time similarly to the above. Terminate after pass K, where $K = \min\{k : |r_i^k - r_i^{k-1}| < \epsilon\}$, $i = 1, \ldots n$, for small enough ϵ. It can be shown that bits will always be pushed right (hence the name FlowRight), and the algorithm terminates in the unique optimal solution[1].

IV. Online Scheduling

We now discuss the realistic problem of online scheduling, *i.e.*, where one does not know future arrivals, channel states, or the arrival rate $\lambda(t)$. The average rate λ is also not known a priori; the online algorithm only knows that $\lambda < \lambda_{\max}$. We assume that the channel statistics are known as well as the present value of the channel gain. We first describe an online scheduling algorithm based on *water-filling in time* that is known to provide optimal adaptation to channel state. Next, we describe the Look-ahead Water-filling algorithm, which simultaneously adapts to both the channel and the backlog.

[1]The proof is omitted due to lack of space.

A. Optimal Online Adaptation to Channel

Recall that any online algorithm needs to ensure stability for $\lambda < \lambda_{\max}$. Therefore, the average rate of transmission (bits/symbol) should be at least $\lambda_{\max}$ (pkts/time unit) $\times$ B(bits/pkt) $\times$ T_s (time units/symbol)[2]. It is known (see [1]) that by optimal adaptation to this ergodic fading channel, the achievable average rate is bounded by the capacity, given by $C = \frac{1}{2}\int_{s_o}^{\infty} \log(\frac{s}{s_o})p(s)d\gamma$ bits/transmission, where $p(s)$ is the probability density of the channel gain s, and s_o is the solution of $\int_{s_o}^{\infty}(\frac{1}{s_o} - \frac{1}{s})p(s)ds = \frac{P}{N}$, where P is the average signal energy per symbol, and N is the noise power. This capacity can be achieved by the following instantaneous transmission power setting $P(s)$, which is the well-known "water-filling in time":

$$P(s) = \begin{cases} \sigma^2(\frac{1}{s_o} - \frac{1}{s}), & \text{if } s \geq s_o \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The required average power and the instantaneous power to be transmitted for the current channel state can be found from the above, by setting $C = \lambda_{\max} \times B \times T_s$.

B. Joint Adaptation to Channel and Backlog

The algorithm presented above optimally adapts to the channel state, assuming that the average rate at which packets must be sent is at least $\lambda_{\max}$. This can be wasteful when the instantaneous packet arrival rate is much lower than $\lambda_{\max}$. Now we shall exhibit an online algorithm that adapts to the arrival rate as well as the channel gain. This online algorithm is based on the Look-Ahead buffer scheme in [9].

The algorithm is as follows: suppose just before time t, a packet transmission ended. Let the backlog at time t be $q(t)$. If $q(t) > 0$, then we begin transmitting the packet at the head of the queue at time t (otherwise, wait until there is a packet in the queue). We set the target transmission rate to $\hat{\mu} = \min\{q(t)/L, \lambda_{\max}\}$ packets/time unit for some constant $L > 0$. Given $\hat{\mu}$, we determine the instantaneous transmission rate according to water-filling. That is, the optimal cutoff value s_o is computed as in section IV-A, which corresponds to an average power for which the capacity is $T_s\hat{\mu}/B$. The current power and rate are then determined from equation (1). We transmit the packet at the head of the queue with this rate.

In the LW algorithm, the target packet transmission rate $\hat{\mu}$ never exceeds $\lambda_{\max}$, yet it can be shown that the queue is stable.

Lemma 2: The LW algorithm is stable, *i.e.*, given any t, with probability one there exists t_1, $t < t_1 < \infty$, such that $q(t_1) = 0$.

[2] Here and in the rest of the paper, for simplicity we assume packets of a constant size of B bits.

To compare the LW algorithm to water-filling, we perform the following experiment : 1 Kbit packets arrive at the buffer at a rate $\lambda < 1$ arrivals/time unit. A time unit is $1/6$ msec, which corresponds to the transmission duration of a packet if it is transmitted at $r = 6$ bits/symbol (symbol rate is constant at 10^6 symbols/sec). The packet arrival process is a Markov Modulated Poisson Process for which $\lambda(t) = \beta\lambda$ with probability $0.9/\beta$, and $\lambda(t) = \frac{1}{10-9/\beta}\lambda$ otherwise, for some $\beta > 0$. The process is ergodic with expected rate λ. Note that when $\beta > 1$, the arrival process is bursty, and for $\beta = 1$ it reduces to a Poisson process at rate λ.

Figure 1 shows an example run of bursty packet arrivals at $\lambda \simeq 0.5$, scheduled by the three algorithms WF (Water-Filling), LW (Look-ahead Water-filling) and OPT (Optimal Offline). Notice that WF transmits with much higher rate than the other two algorithms, thus quickly finishes its backlog and idles a significant amount of the time. LW, on the other hand, spreads its rate more uniformly over time, almost as uniformly as OPT which has the lowest rate transmission.

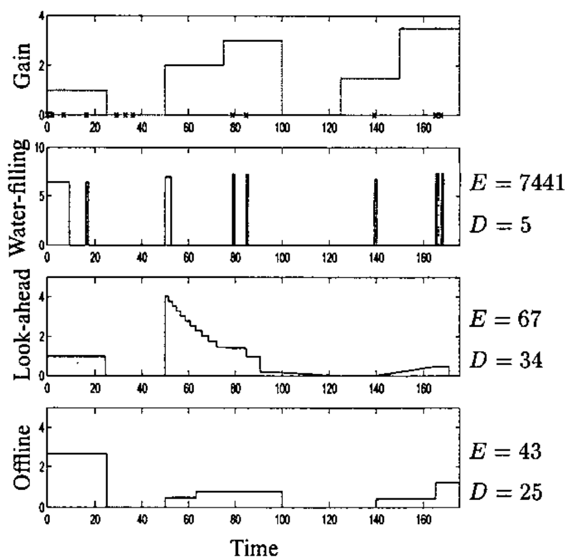

Fig. 1. The top plot shows a sequence of packet arrivals ("×") and channel gains. The lower three plots show the instantaneous rates used by the online algorithms Water-filling and Look-ahead Water-filling, and the Optimal Offline Schedule, respectively, as they run on this sequence of packet arrivals. The average energy per packet values are normalized for a noise power of unity.

Figures 2 and 3 explore the energy and delay performance of these algorithms. Note that the water-filling schedule has constant energy for all arrival rates, since the rate it assigns to packets is independent of λ. This energy is much higher than the average energy values achieved by LW when λ is small; both for bursty and non-bursty arrival processes. This energy efficiency is achieved at the expense of an increase in delay. The delay of LW is around L, as it essentially uses this much time to "look ahead" and monitor the arrival process: Whenever the queue empties, the first packet that arrives is transmitted for a duration L while packets that follow it are queued.

However, as can be observed from Figure 2, the variation in the delay of LW is much smaller than that of WF. In the figure, the delay of WF varies by about 7000% as λ is varied from 0 to $\lambda_{\max} = 1$, while the delay of LW varies only about 60%. The fact that the *delay jitter* is so small makes the backlog-adaptive algorithm attractive for data applications, especially streaming media.

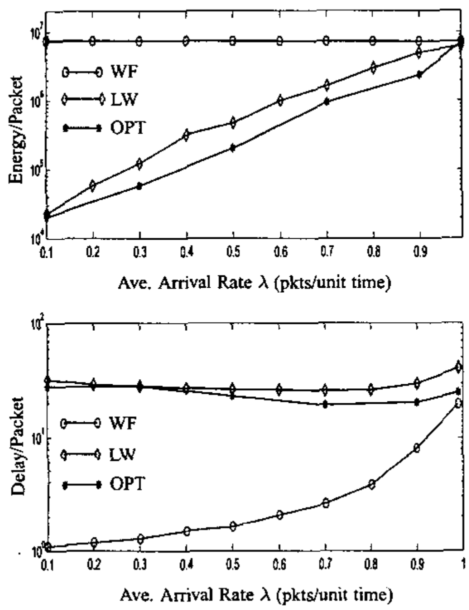

Fig. 2. Energy per packet as arrival rate changes for $\beta = 1$ in Rayleigh fading; $L = 25$.

V. Extension to Multi-Access Channels with Fading

The single-user offline and online scheduling results presented thus far can be extended to fading multiple-access and broadcast channels. To formulate the offline scheduling problem we first merge all users' packet arrival sequences and the times at which channel states change to obtain m epochs and note as before that in an optimal schedule rates do not need to change during an epoch. The problem is then:

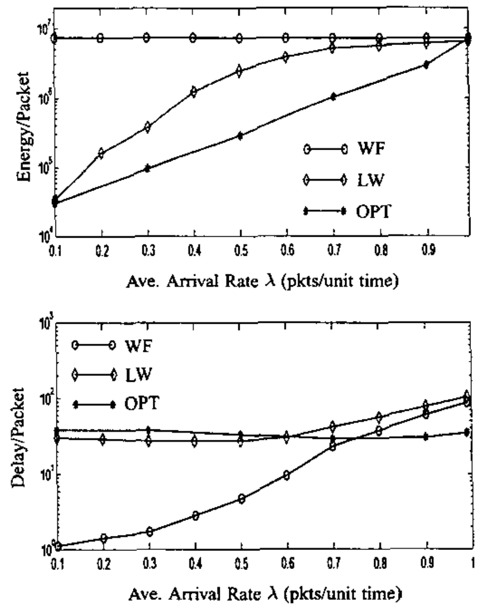

Fig. 3. Average energy per packet as arrival rate changes for $\beta = 2$ in Rayleigh fading; $L = 25$.

Minimize:

$$\sum_{i=1}^{m} \xi_i \left(\left(\frac{1}{s_{k_1 i}} - \frac{1}{s_{k_2 i}}\right) f(r_{k_1 i}) + \frac{1}{s_{k_2 i}} f(r_{k_1 i} + r_{k_2 i}) \right)$$

subject to:

$$\sum_{i=1}^{k} r_{ji}\xi_i \le \sum_{i=1}^{k} c_{ji} \quad k = 1, \ldots, m-1, \;\; j = 1,2$$

$$\sum_{i=1}^{m} r_{ji}\xi_i = \sum_{i=1}^{m} c_{ji} \quad j = 1,2,$$

where $k_1 i = \arg\min_{k\in\{1,2\}}(s_{1i}, s_{2i})$ and $k_2 i = \{1,2\} - \{k_1 i\}$, and where $c_{ji} = B$ if a packet for user j arrives at the beginning of epoch i, and 0 otherwise. This is a convex optimization problem with linear constraints and can be solved by FlowRight.

Recall that in the single-user case, the optimal adaptation to the channel state is given by the water-filling solution. In the multiple user case, analogous results exist. When the fading processes of users are i.i.d., and the goal is to maximize the sum rate with respect to a total power constraint, the important result of Knopp and Humblet [2] says that the optimal power control scheme allows only the user with the best channel to transmit at any given time. The rate of that user is then determined by water-filling across the channel states. Tse and Hanly [3] exhibit the optimal power control when users are not necessarily symmetric, and the goal is to maximize a weighted sum of the rates. They propose a "Greedy Algorithm" which also solves the dual problem, *i.e.*, achieves a given vector of average rates with minimum power. The Greedy algorithm is an optimal online algorithm, as long as the transmitter knows the fading state.

A "look-ahead greedy" online schedule that uses a look-ahead buffer to adapt to both backlog and channel state (similar to Look-ahead Water-filling) can be obtained as follows: Each user's required rate is estimated from the current backlogs. The power allocation is then determined using the Greedy algorithm in [3].

Finally, we note that the broadcast scheduling problem in the slow fading channel can be stated and solved similarly.

VI. Conclusion

Schedules such as water-filling in time that optimally adapt to the state of the fading channel can be energy-inefficient in the case of variable-rate data. This paper considered minimum-energy scheduling by jointly adapting to the channel state and the rate of data generation while keeping delay bounded. The performance of the Look-ahead Water-filling algorithm is close to that of the optimal offline schedule. The algorithm essentially makes use of the diversity in the data rate and adapts the transmission rate to the number of packets in the transmitter buffer, thereby avoiding unnecessary high-rate transmissions and idle periods when the instantaneous data rates are low. This heuristic algorithm can be approximated by practical coding/modulation schemes. Finally, results can be generalized to fading multi-access and broadcast channels.

Acknowledgments: The authors would like to thank Mayank Sharma and Sina Zahedi for their valuable suggestions. The work was partially supported under SNRC Multilayer Mobile Networking Project and partly by a Texas Instruments Stanford Graduate Fellowship.

References

[1] A. Goldsmith, "The capacity of downlink fading channels with variable rate and power," *IEEE Trans. on Vehicular Technology*, vol. 46, no. 3, August 1997.

[2] R. Knopp and P.A. Humblet, "Information capacity and power control in single-cell multiuser communications," presented at *Intl. Conference on Communications*, Seattle, WA, June 1995.

[3] D. Tse and S. Hanly, "Multi-access fading channels: part I: polymatroid structure, optimal resource allocation and throughput capacities", *IEEE Transactions on Information Theory*, vol. 44, No. 7, pp. 2796-2815, Nov., 1998.

[4] A. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Trans. on Communication*, vol. 45, pp. 1218-1230, Oct. 1997.

[5] K.J. Hole, H. Holm, and G.E. Øien, "Adaptive multidimensional coded modulation over flat fading channels," *IEEE Journal on Selected Areas in Comm.*, vol. 18, no. 7, July 2000.

[6] E. Uysal-Biyikoglu, B. Prabhakar and A. El Gamal, "Energy-efficient packet transmission over a wireless link," in *IEEE/ACM Trans. on Networking*, vol. 10, August 2002.

[7] E. Uysal-Biyikoglu and A. El Gamal, "Energy-efficient packet transmission over a multi-access channel," in *Proc. 2002 Intl. Symposium on Information Theory*, 2002, p. 153.

[8] W.S. Yoon and T.E. Klein, "Delay-optimal power control for wireless data users with average power constraints," in *Proc. 2002 Int. Symposium on Information Theory*, 2002, p. 53.

[9] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks", in *Proc. IEEE INFOCOM*, 2002, pp. 1773-1782.