

Delay Optimal Power Aware Opportunistic Scheduling with Mutual Information Accumulation

Xiaohan Wei and Michael J. Neely

Abstract—This paper considers optimization of power and delay in a time-varying wireless link using rateless codes. The link serves a sequence of variable-length packets. Each packet is coded and transmitted over multiple slots. Channel conditions can change from slot to slot and are unknown to the transmitter. The amount of mutual information accumulated on each slot depends on the random channel realization and the power used. The goal is to minimize average service delay subject to an average power constraint. We formulate this problem as a frame-based stochastic optimization problem and solve it via an online algorithm. We show that the subproblem within each frame is a simple integer program which can be effectively solved using a dynamic program. The optimality of this online algorithm is proved using the frame-based Lyapunov drift analysis.

I. INTRODUCTION

Consider a slotted time system where a single source transmits a sequence of data packets to a receiver over a time varying channel. During each time slot, the source transmits a portion of the mutual information needed for the receiver to decode the current packet. This is done by choosing the amount of power on each transmission. The source does not know the current channel state when it makes this decision. The chosen power and the random channel state determine the amount of mutual information that is received. The transmission of that packet is finished when the receiver accumulates enough mutual information to decode. This technique is usually referred to as mutual information accumulation ([1]-[2]) and can be implemented through rateless coding schemes such as Raptor codes and Fountain codes ([3]-[4]).

Our goal is to design a transmission algorithm such that the time average delay of the sequence of packets is minimized and a time average power constraint is satisfied. Specifically, we consider an online setting where both the channel conditions and the packet lengths could vary according to some underlying probability distributions. This problem is challenging for two reasons: (i) Decoding times can vary randomly for each packet, (ii) The power spent during the current time slot affects the desired power constraint and also influences how much information is left to transmit for that packet in future slots, creating complex dependencies over different slots. Fortunately, the system contains an underlying Markov renewal structure that is exploited in this paper.

In this paper, we use renewal reward theory (e.g. [5]) together with Lyapunov optimization (e.g. [6]) to develop an online algorithm for this problem. Previous work [7] introduces

a general framework for applying Lyapunov optimization to renewal systems through the idea of dividing the time line into renewal frames and solving a subproblem within each renewal frame. Works in [8] and [9] analyze the performance of frame-based Lyapunov optimization for a file downloading problem and a multiclass M/G/1 queue problem, respectively. In both papers, the subproblem within each frame is effectively solved, which is the key to the success of the whole algorithm. In this paper, we show that the subproblem within each frame is an integer program which can be effectively solved using a dynamic program.

Prior work on mutual information accumulation in [10][11][2][12] focus on a network with fixed channel conditions. Works in [10] and [11] consider the problem of minimum energy accumulative broadcast and show that it is NP-complete. Works [2] and [12] consider the problem of optimal cooperative routing in a multi-hop network and develop linear program based formulations. Applications to time-varying channel conditions are considered in [13] and [14]. All these previous works only consider the optimal transmission of a single packet. Our work differs from these works in that we consider a sequential transmission scenario with a time average power constraint.

The contributions of the paper are as follows: First, we formulate this power constrained minimum delay problem as a frame-based stochastic optimization problem, where each frame is the time period for transmitting one single packet. Second, we propose a provably optimal frame-based online algorithm where an integer program is solved during each frame. We show that the integer program can be effectively solved through dynamic programming. A nice feature of the resulting algorithm is that, while it assumes knowledge of the channel probability distribution, it does not require knowledge of the file size probability distribution. Further, the power constraint is shown to hold even if there is a mismatch between the actual channel probability distribution and the assumed distribution.

II. PROBLEM FORMULATION

In this section, we present our system model and the delay minimization problem. Consider a slotted time single hop network where one source is transmitting a sequence of packets to a receiver with time varying channel states. The transmitter uses a rateless coding scheme where the receiver accumulates mutual information about each packet until it can successfully decode. The goal is to minimize the time average packet delay subject to an average power

The authors are with the Electrical Engineering department at the University of Southern California, Los Angeles, CA.

This material is supported in part by the NSF grant 0964479.

constraint on the source. For simplicity of analysis, we assume that the source's only significant power expenditure lies in transmissions, and coding/decoding yields negligible power and time consumption.

Let $\delta > 0$ be a fixed amount of mutual information, called a *data unit*. Let $L[f]$ be the length of the f -th packet, which is a positive integer number of data units associated with packet f . We assume that $L[f]$ is i.i.d. over $f \in \{0, 1, 2, \dots\}$ and takes values in a finite set \mathcal{L} . Also, assume that the source uses *ideal rateless codes* for transmission so that the mutual information collected by the receiver from different slots for the same packet add up with each other. Then, the receiver can decode packet f if and only if it accumulates at least $L[f]$ data units for that packet. Finally, we assume that the source cannot transmit any new packet until the transmission of the previous packet is finished.

The source has different power spectral density (PSD) options to take, which is in units of joules/slot/Hz. Let $P(t) \in \mathcal{P}$ be the option taken by the source at slot t . We assume $\mathcal{P} = \{P_1, P_2, \dots, P_{|\mathcal{P}|}\}$ is finite. Without loss of generality, further assume $0 < P_1 < P_2 < \dots < P_{|\mathcal{P}|}$. The channel power gain at time slot t is denoted as $\alpha(t)$. We assume $\alpha(t) > 0$ is i.i.d. over slots and takes values in a finite set $\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_{|\mathcal{A}|}\}$ with probability distribution $\phi(\alpha_i)$. They satisfy

$$\phi(\alpha_i) \geq 0, \quad \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) = 1.$$

Then, the mutual information transmitted during a single time slot t can be expressed as $K(\alpha(t), P(t))$, which is also a positive integer number of data units. We make no assumption on the specific form of $K(\alpha(t), P(t))$ except that it must be nondecreasing with respect to both $\alpha(t)$ and $P(t)$. For example, if the source uses fountain codes, then by Shannon's formula, $K(\alpha(t), P(t)) = \left\lceil \log_2 \left(1 + \frac{\alpha(t)P(t)}{N_0} \right) \right\rceil$, where $N_0/2$ denotes the PSD of the white noise process.

For the ease of algorithm development, we further segment the slotted timeline into *frames*. Define frame f as the time period (in units of slots) of transmitting packet f , with starting slot denoted as t_f and frame length denoted as $T[f] \triangleq t_{f+1} - t_f$. Frame f ends on the first slot in which the total mutual information for that frame exceeds $L[f]$, which is

$$\sum_{t=t_f}^{t_{f+1}-1} K(\alpha(t), P(t)) \geq L[f].$$

We make the following assumption regarding the random processes $\{L[f]\}_{f=0}^{\infty}$ and $\{\alpha(t)\}_{t=0}^{\infty}$:

Assumption 1. *The distribution of packet length process $\{L[f]\}_{f=0}^{\infty}$ is unknown to the source but the source can observe $L[f]$ at the beginning of each frame. The process $\{\alpha(t)\}_{t=0}^{\infty}$ is unobservable, meaning that the source cannot see $\alpha(t)$ before transmission at slot t , but the source has the knowledge of distribution $\phi(\alpha_i)$, $\alpha_i \in \mathcal{A}$.*

Furthermore, at the end of slot t , the receiver sends a feedback signal so that the source knows how much mutual

information has been delivered. The goal is to choose the PSD allocation $P(t)$ at each time slot so as to minimize the average delay (in units of slots/packet) subject to a time average power constraint:

$$\begin{aligned} \min \quad & \lim_{F \rightarrow \infty} \frac{\sum_{f=0}^{F-1} T[f]}{F} \\ \text{s.t.} \quad & \lim_{F \rightarrow \infty} \frac{\sum_{f=0}^{F-1} \sum_{t=t_f}^{t_{f+1}-1} P(t)}{\sum_{f=0}^{F-1} T[f]} \leq \beta, \end{aligned}$$

where β is a positive constant. An equivalent formulation which does not include a ratio of time averages in the constraint is as follows:

$$\min \quad \lim_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} T[f] \quad (1)$$

$$\text{s.t.} \quad \lim_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \leq 0. \quad (2)$$

For the rest of the paper, we shall consider (1)-(2) instead. We also need the assumption that the minimal PSD allocation P_1 is greater than 0 but small enough. This ensures that the frame length $T[f]$ is finite for any policy, and the problem described is feasible.

Assumption 2. *The minimal PSD option P_1 satisfies $0 < P_1 < \beta$.*

Furthermore, let

$$\begin{aligned} L_{\max} &= \max_{L \in \mathcal{L}} L, \\ K_{\min} &= \min_{\alpha \in \mathcal{A}} K(\alpha, P_1). \end{aligned}$$

Here L_{\max} stands for maximum packet length and K_{\min} stands for minimum per slot mutual information transmission. Notice that since $K(\alpha, P)$ is a positive integer for any α and P , we have $K_{\min} \geq 1$.

III. AN ONLINE ALGORITHM

In this section, we propose an online algorithm which adapts to the time varying packet lengths and channel power gains.

A. Proposed algorithm

The proposed algorithm operates over frames, and features a "virtual queue" technique treating the time average power constraint. Define a virtual queue $Q[f]$, which has initial condition $Q[0] = 0$ and update equation:

$$Q[f+1] = \max \left\{ Q[f] + \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta), 0 \right\}. \quad (3)$$

The proposed algorithm then runs in the following two steps with a trade-off parameter $V > 0$:

- At the beginning of frame f , observe $Q[f]$, $L[f]$, and choose a sequence of PSD allocations so as to solve the following optimization problem:

$$\min_{P(t) \in \mathcal{P}} \mathbb{E} \left[\sum_{t=t_f}^{t_{f+1}-1} R_f(P(t)) \middle| Q[f], L[f] \right] \quad (4)$$

$$\text{s.t.} \quad \sum_{t=t_f}^{t_{f+1}-1} K(\alpha(t), P(t)) \geq L[f] \quad (5)$$

where for any allocation $P(t) \in \mathcal{P}$, the function $R_f(P(t))$ is defined as $R_f(P(t)) \triangleq V + Q[f](P(t) - \beta)$, and the expectation is taken with respect to the random channel gain $\alpha(t)$.

- Update $Q[f]$ according to (3) at the end of each frame.

Section IV shows the above algorithm structure produces a near optimal solution, where proximity to optimality is determined by the V parameter. The following section shows how to implement each state of the algorithm, that is, how to solve the problem (4)-(5).

B. Solve (4)-(5) via dynamic programming under static channel gain

In this section, we show how to solve (4)-(5) during each frame under static channel gain, i.e. $\alpha(t) = \alpha_0$, $\forall t$, where α_0 is a positive constant. Then, we show how to generalize the algorithm to random channel case in the next section.

Given $Q[f]$ and $L[f]$, there is no randomness in the formulation (4)-(5). Thus, we can simplify it as follows:

$$\min_{P(t) \in \mathcal{P}} \sum_{t=t_f}^{t_{f+1}-1} R_f(P(t)) \quad (6)$$

$$\text{s.t.} \quad \sum_{t=t_f}^{t_{f+1}-1} K(\alpha_0, P(t)) \geq L[f]. \quad (7)$$

Next, we distinguish between the following two cases:

- 1) Case 1: Suppose there exists some option $P_{j_0} \in \mathcal{P}$ such that $R_f(P_{j_0}) < 0$, then, since both $R_f(P(t))$ and $K(\alpha_0, P(t))$ are increasing with $P(t)$, the best decision is always choosing the smallest power allocation (i.e. $P(t) = P_1$) till the end of the frame.
- 2) Case 2: Suppose $R_f(P_j) \geq 0$, $\forall P_j \in \mathcal{P}$. Then, we can interpret above problem in the following way: For each PSD option $P_j \in \mathcal{P}$, there is a corresponding penalty $R_f(P_j)$ and a weight $K(\alpha_0, P_j)$. The goal is to minimize the total penalty subject to the constraint that the total weight should be at least $L[f]$. Since there are only finite options in \mathcal{P} , (6)-(7) can be written as the following integer program problem:

$$\begin{aligned} \min_{x_j, j \in \{1, \dots, |\mathcal{P}|\}} & \sum_{j=1}^{|\mathcal{P}|} R_f(P_j) x_j \\ \text{s.t.} & \sum_{j=1}^{|\mathcal{P}|} K(\alpha_0, P_j) x_j \geq L[f], \quad x_j \in \mathbb{N}. \end{aligned}$$

Notice that both $K(\alpha_0, P_j)$ and $L[f]$ are positive integers. This is similar to a classical Knapsack problem. A standard technique for solving this type of problems is dynamic programming (see [15] for related results). Define the state variable k , $k \in \mathbb{Z}$ and define $m[k]$, $k \in \mathbb{Z}$ as the minimum value achieved by solving the following problem:

$$\min_{x_j, j \in \{1, \dots, |\mathcal{P}|\}} \sum_{i=1}^{|\mathcal{P}|} R_f(P_j) x_j \quad (8)$$

$$\text{s.t.} \quad \sum_{i=1}^{|\mathcal{P}|} K(\alpha_0, P_j) x_j \geq k, \quad x_j \in \mathbb{N}. \quad (9)$$

Define $m[k] = 0$, $\forall k \leq 0$, then, we progressively compute $m[k]$ for each $k \in \{1, 2, \dots, L[f]\}$ as follows:

$$\begin{aligned} m[k] &= \min_{j \in \{1, 2, \dots, |\mathcal{P}|\}} R_f(P_j) + m[k - K(\alpha_0, P_j)], \\ &\forall k > 0. \end{aligned}$$

The proof of optimality is similar to that of knapsack problem which can be found, for example, in chapter 6 of [15].

C. Extension to random channel gain

In this section, we solve (4)-(5) under random channel gains. Again, we distinguish between the following two cases:

- 1) Suppose there exists some option $P_{j_0} \in \mathcal{P}$ such that $R_f(P_{j_0}) < 0$. Then, since both $R_f(P(t))$ and $K(\alpha(t), P(t))$ are increasing with $P(t)$ for any $\alpha(t) \in \mathcal{A}$, the best decision is always choosing the smallest power allocation (i.e. $P(t) = P_1$) until the end of the frame.
- 2) Suppose $R_f(P_j) \geq 0$, $\forall P_j \in \mathcal{P}$. Then, in a similar way as before, we get a randomized integer program, which can still be solved by dynamic program. Define k , $k \in \mathbb{Z}$ as the state variable and define $m[k]$, $k \in \mathbb{Z}$ as the minimum value achieved by the following problem:

$$\min_{P(t) \in \mathcal{P}} \mathbb{E} \left[VT[f] + Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \middle| Q[f], L[f] \right] \quad (10)$$

$$\text{s.t.} \quad \sum_{t=t_f}^{t_{f+1}-1} K(P(t), \alpha(t)) \geq k \quad (11)$$

Define $m[k] = 0$, $\forall k \leq 0$, then, we progressively compute $m[k]$ for each $k \in \{1, 2, \dots, L[f]\}$ as follows

$$\begin{aligned} m[k] &= \min_{j \in \{1, 2, \dots, |\mathcal{P}|\}} R_f(P_j) + \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) m[k - K(\alpha_i, P_j)], \end{aligned} \quad (12)$$

The proof that (12) indeed leads to optimal policy during each frame is similar to that of a standard proof of optimality for a dynamic program (see [16] for more details). For completeness, we provide a proof using induction in Appendix A.

IV. PERFORMANCE ANALYSIS

In this section, we prove that the online algorithm in Section III-A satisfies the power constraint and achieves the near optimal throughput (with $\mathcal{O}(1/V)$ optimality gap). We first show that the virtual queue $Q[f]$ is bounded, which implies that the time average of “arrival process” $\sum_{t=t_f}^{t_{f+1}-1} P(t)$ is less than or equal to that of “service process” $\beta T[f]$, and thus the power constraint is satisfied. Then, we prove that the algorithm gives near optimal performance by Lyapunov drift analysis.

A. Average power constraint via queue bound

Lemma 1. *If there exists a constant $C > 0$ such that $Q[f] \leq C$, then, it follows for any $F > 0$,*

$$\frac{1}{F} \sum_{f=0}^{F-1} \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \leq \frac{C}{F},$$

thus,

$$\limsup_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \leq 0.$$

Proof. From (3), we know that for each frame f :

$$Q[f+1] \geq Q[f] + \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta)$$

Rearranging terms gives:

$$\sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \leq Q[f+1] - Q[f]$$

Fix $F > 0$. Summing over $f \in \{0, 1, \dots, F-1\}$ gives:

$$\sum_{f=0}^{F-1} \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \leq Q[F] - Q[0] \leq C,$$

which implies:

$$\frac{1}{F} \sum_{f=0}^{F-1} \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \leq \frac{C}{F}.$$

Taking $F \rightarrow \infty$ gives the desired result. \square

The next lemma shows that the queue process deduced from proposed algorithm is deterministically bounded.

Lemma 2. *If $Q[0] = 0$, then, the proposed algorithm gives for all $f \in \mathbb{N}^+$,*

$$Q[f] \leq \max \left\{ \frac{V}{\beta - P_1} + \left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil \cdot (P_{|\mathcal{P}|} - \beta), 0 \right\},$$

where $\lceil x \rceil$ denotes the ceiling function rounding up towards the smallest integer greater than x .

Proof. First, consider the case when $P_{|\mathcal{P}|} \leq \beta$. From (3) it is obvious that $Q[f]$ can never increase. Thus, $Q[f] = 0, \forall f \in \mathbb{N}^+$.

Next, consider the case when $P_{|\mathcal{P}|} > \beta$. It is obvious that the frame length $T[f]$ is upper bounded by $\left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil$. We then

prove the assertion by induction on f . The result trivially holds for $f = 0$. Suppose it holds at $f = l$ for $l \in \mathbb{N}^+$, so that:

$$Q[l] \leq \frac{V}{\beta - P_1} + \left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil \cdot (P_{|\mathcal{P}|} - \beta).$$

We are going to prove the same holds for $f = l + 1$. There are two further cases to be considered:

1) $Q[l] \leq \frac{V}{\beta - P_1}$. In this case we have by (3):

$$\begin{aligned} Q[l+1] &\leq Q[l] + T[l](P_{|\mathcal{P}|} - \beta) \\ &\leq \frac{V}{\beta - P_1} + \left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil \cdot (P_{|\mathcal{P}|} - \beta). \end{aligned}$$

2) $\frac{V}{\beta - P_1} < Q[l] \leq \frac{V}{\beta - P_1} + \left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil \cdot (P_{|\mathcal{P}|} - \beta)$. In this case, we have by first half of induction hypothesis,

$$Q[l]P_1 + V - \beta Q[l] < 0.$$

By the first case in Section III-C, the source will choose power allocation $P(t) = P_1, \forall t \in [t_l, t_{l+1} - 1]$. Since $P_1 < \beta$ by assumption 2, it follows from the queue updating rule (3) that the queue will increase no more at the end of frame l . By the second half of the induction hypothesis,

$$Q[l+1] \leq \frac{V}{\beta - P_1} + \left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil \cdot (P_{|\mathcal{P}|} - \beta).$$

This finishes the proof. \square

Combining above two lemmas gives the result that the proposed algorithm satisfies the power constraint. Notice that above result is a sample path result which only assumes $K_{\min} > 0, \beta - P_1 > 0$ and probability $\phi(\cdot) \in [0, 1]$. Thus, the algorithm meets the average power constraints even if it uses incorrect values for these parameters.

B. Optimization over randomized stationary algorithm

Suppose the server knows the probability distribution of packet length. Then, consider the following class of algorithm: At the beginning of each frame f , after observing the packet length $L[f]$, the source independently selects a sequence of power allocation $P^*(t), t \in [t_f, t_{f+1} - 1]$ according to some pre-specified probability distribution which depends only on $L[f]$ and feedback during that frame.

Then, since $L[f]$ is an i.i.d. process over frames, this algorithm is stationary. Furthermore, let $\{T^*[f]\}_{f=0}^{\infty}$ and $\left\{ \sum_{t=t_f}^{t_f+T^*[f]-1} P^*(t) \right\}_{f=0}^{\infty}$ be two processes corresponding to this randomized algorithm. It can be shown that these two processes are both i.i.d. over frames. By strong law of large numbers,

$$\lim_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} T^*[f] = \mathbb{E}[T^*[f]], \quad w.p.1,$$

$$\begin{aligned} &\lim_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} \sum_{t=t_f}^{t_f+T^*[f]-1} (P^*(t) - \beta) \\ &= \mathbb{E} \left[\sum_{t=t_f}^{t_f+T^*[f]-1} (P^*(t) - \beta) \right], \quad w.p.1. \end{aligned}$$

Furthermore, it can be shown that the optimal solution to (1)-(2) can be achieved over this class of algorithms (see [7] for related results). That being said, there exists a randomized stationary algorithm such that

$$\mathbb{E}[T^*[f]] = \theta^*, \quad (13)$$

$$\mathbb{E} \left[\sum_{t=t_f}^{t_f+T^*[f]-1} (P^*(t) - \beta) \right] \leq 0, \quad (14)$$

where θ^* is the optimal delay solving (1)-(2). As we shall see, this result plays an important role in our subsequent proof of optimality. Unfortunately, this algorithm is not implementable because the source has no knowledge on the distribution of packet length $L[f]$. Even if it has the knowledge, the computation complexity is very high because of the possibly large packet length set \mathcal{L} .

C. Key feature inequality

We give a feature of our proposed online algorithm which is the key to the proof of optimality. Define $\mathcal{H}[f]$ as the *system history* up to frame f . Consider the algorithm that at the beginning of frame f , observes $Q[f]$, $L[f]$ and then chooses a sequence of PSD allocations $P(t) \in \mathcal{P}$ so as to minimize (4)-(5). Then, the policy must have achieved a smaller value on (4) compared to that of best stationary policy from Section IV-B. Mathematically, this can be expressed as the following inequality:

$$\begin{aligned} & \mathbb{E} \left[VT[f] + Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \middle| \mathcal{H}[f], L[f] \right] \\ & \leq \mathbb{E} \left[VT^*[f] + Q[f] \sum_{t=t_f}^{t_f+T^*[f]-1} (P^*(t) - \beta) \middle| \mathcal{H}[f], L[f] \right], \end{aligned}$$

where the left-hand-side uses values $T[f]$ and $P(t)$ from the implemented policy, and the right-hand-side uses values $T^*[f]$ and $P^*(t)$ from the stationary policy.

Taking expectation from both sides regarding $L[f]$ gives

$$\begin{aligned} & \mathbb{E} \left[VT[f] + Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \middle| \mathcal{H}[f] \right] \\ & \leq \mathbb{E} \left[VT^*[f] + Q[f] \sum_{t=t_f}^{t_f+T^*[f]-1} (P^*(t) - \beta) \middle| \mathcal{H}[f] \right]. \end{aligned}$$

Using the fact that the stationary randomized policy is independent of history as well as (13) and (14),

$$\mathbb{E} \left[VT[f] + Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \middle| \mathcal{H}[f] \right] \leq V\theta^* \quad (15)$$

D. Near optimal performance

Theorem 1. *The proposed algorithm satisfies the constraint and yields the time average delay satisfying*

$$\limsup_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} T[f] \leq \theta^* + \frac{C_0}{V},$$

with probability 1, where C_0 is a positive constant independent of V .

Proof. First, it is obvious from the two lemmas in Section IV-A that the time average constraint is satisfied. The rest is devoted to proving the delay guarantee.

Define the Lyapunov drift $\Delta[f]$ for the virtual queue $Q[f]$ as follows

$$\Delta[f] = \frac{1}{2} (Q[f+1]^2 - Q[f]^2).$$

Substitute the definition of $Q[f+1]$ gives

$$\begin{aligned} \Delta[f] & \leq \frac{1}{2} \left(Q[f] + \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \right)^2 - \frac{1}{2} Q[f]^2 \\ & = \frac{1}{2} \left(\sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) \right)^2 + Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta). \end{aligned}$$

Recall that the frame length $T[f]$ is upper bounded by $\left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil$ and $P(t) \leq P_{|\mathcal{P}|}$. Let $C_0 \triangleq \frac{1}{2} \left(\left\lceil \frac{L_{\max}}{K_{\min}} \right\rceil (P_{|\mathcal{P}|} + \beta) \right)^2$, it follows

$$\Delta[f] \leq C_0 + Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta).$$

Now, we consider the following expected drift term $\Delta[f]$ plus penalty term $VT[f]$:

$$\begin{aligned} & \mathbb{E}[\Delta[f] + VT[f] | \mathcal{H}[f]] \\ & \leq C_0 + \mathbb{E} \left[Q[f] \sum_{t=t_f}^{t_{f+1}-1} (P(t) - \beta) + VT[f] \middle| \mathcal{H}[f] \right] \\ & \leq C_0 + V\theta^*, \end{aligned}$$

where the last inequality follows from (15). Thus, rearranging the terms gives

$$\mathbb{E}[\Delta[f] + V(T[f] - \theta^*) | \mathcal{H}[f]] \leq C_0. \quad (16)$$

Since $Q[f]$ is deterministically bounded, $\Delta[f]$ is also deterministically bounded, thus, the following holds

$$\sum_{f=0}^{\infty} \frac{\mathbb{E}[\Delta[f]^2]}{f^2} < \infty. \quad (17)$$

Corollary 4.2 of [17] states that if $\Delta[f]$ satisfies (17), then, (16) implies the following holds with probability 1:

$$\limsup_{F \rightarrow \infty} \frac{1}{F} \sum_{f=0}^{F-1} V(T[f] - \theta^*) \leq C_0.$$

Divide both sides by V and rearrange the terms gives the desired result. \square

V. SIMULATION EXPERIMENTS

In this section, we demonstrate the near optimality of the proposed algorithm by simulations. We show that as the trade-off parameter V gets larger, the suboptimality gap gets smaller, but in turn the average virtual queue size, which affects the convergence rate of the algorithm (as is shown in Lemma 1), also gets larger.

We consider a scenario where the information packet length is drawn independently and identically from the set $\mathcal{L} = \{10, 20, 30\}$ with probability 0.2, 0.3 and 0.5, respectively. The PSD option set is $\mathcal{P} = \{1, 2, 4, 8\}$ and the power constraint $\beta = 3$. The channel power gain takes values in the set $\mathcal{A} = \{1, 2\}$ with equal probability. Finally, we assume the mutual information is transmitted according to Shannon's formula: $K(\alpha(t), P(t)) = \left\lceil \log_2 \left(1 + \frac{\alpha(t)P(t)}{N_0} \right) \right\rceil$ during each time slot t , with white noise PSD $N_0 = 1$. The algorithm is run for 1 million slots on each V value. We compare the performance of our proposed algorithm with the optimal randomized stationary policy. The optimal policy is computed using a linear program with the knowledge of the packet length distribution. The details of formulating a Markov decision process into a linear program is shown in [18].

In Fig. 1, we show that as our tradeoff parameter V gets larger, the average delay of the proposed algorithm approaches the optimal delay. Fig.2 and Fig. 3 show that V also affects the constraint gap and the average virtual queue size. As V gets larger, the average queue size becomes larger and the constraint gap becomes smaller. The plot of the theoretical upper bound of the virtual queue size from Lemma 2, which demonstrates that the virtual queue is deterministically bounded.

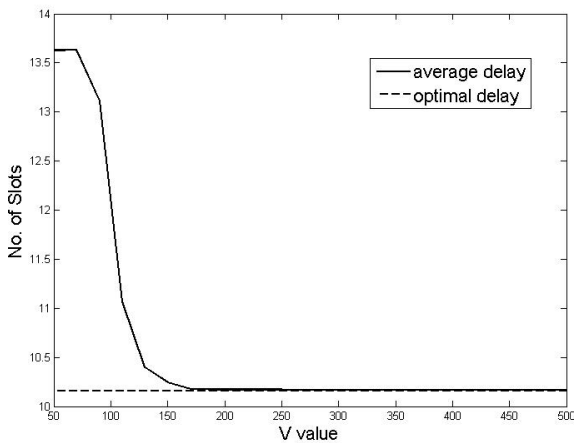


Fig. 1. Time average delay versus V parameter over 1 million slots.

APPENDIX A

PROOF OF OPTIMALITY FOR (12)

Proof. First of all, for any $k \leq 0$, since $R_f(P_j) \geq 0, \forall j \in \{1, 2, \dots, |\mathcal{P}|\}$, the best solution is not choosing any PSD allocation at all. This results in $m[k] = 0, \forall k \leq 0$. Next, for $k > 0$, we use induction to show $m[k]$ indeed satisfies the iteration (12).

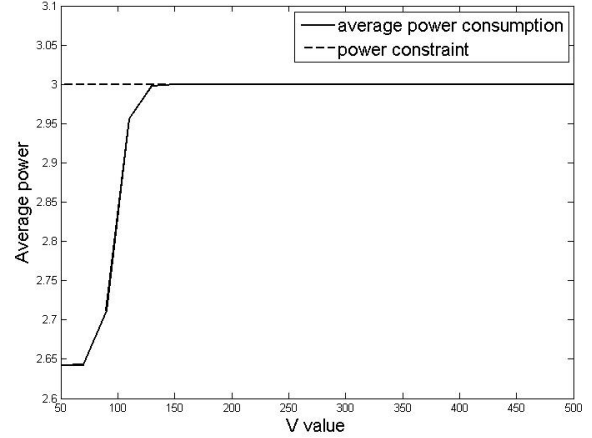


Fig. 2. Time average power versus V parameter over 1 million slots.

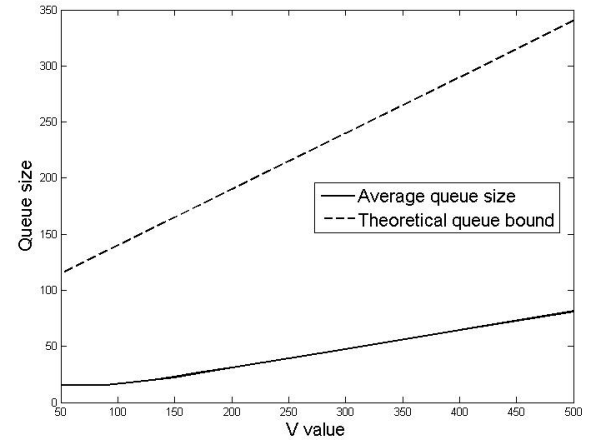


Fig. 3. Time average queue size versus V parameter over 1 million slots.

- 1) Base case $k = 1$. Since $K(\alpha_i, P_j)$ is a positive integer, (12) gives $m[1] = \min_j R(P_j)$. On the other hand, the best solution is to place one weight which has the least penalty $R(P_j)$, which obviously implies $m[1]$ satisfies the iteration (12).
- 2) Suppose $m[k]$ satisfies (12) for all $k \in \{1, 2, \dots, k_0\}$, then, we want to prove $m[k_0 + 1]$ also satisfies (12). We show that

$$m[k_0 + 1] \leq \min_{j \in \{1, \dots, |\mathcal{P}|\}} R_f(P_j) + \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) m[k_0 + 1 - K(\alpha_i, P_j)] \quad (18)$$

and

$$m[k_0 + 1] \geq \min_{j \in \{1, \dots, |\mathcal{P}|\}} R_f(P_j) + \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) m[k_0 + 1 - K(\alpha_i, P_j)] \quad (19)$$

respectively.

Proof of (18): Consider the following specific allocation policy: Pick any $P_j \in \mathcal{P}$, and then adopt the policy which solves (10)-(11) for $k = k_0 + 1 - K(\alpha_i, P_j)$. By induction hypothesis, the resulting penalty of this policy is $R_f(P_j) + \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) m[k_0 + 1 - K(\alpha_i, P_j)]$. Since this is a specific policy, it follows,

$$m[k_0 + 1] \leq R_f(P_j) + \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) m[k_0 + 1 - K(\alpha_i, P_j)],$$

$$\forall j \in \{1, \dots, |\mathcal{P}|\},$$

which implies (18).

Proof of (19): Suppose $\hat{P}(t_f), \hat{P}(t_f+1), \dots, \hat{P}(t_{f+1}-1)$ is a sequence of PSD allocations generated by the optimal policy minimizing (10)-(11) for $k = k_0 + 1$. Given the event $\alpha(t_f)$ and allocation $\hat{P}(t_f)$ at time slot t_f , we look at the remaining sequence of PSD allocations $\hat{P}(t_f + 1), \dots, \hat{P}(t_{f+1} - 1)$. It can be regarded as a sequence generated by a specific policy under the constraint

$$\sum_{t=t_f+1}^{t_{f+1}-1} K(\alpha(t), \hat{P}(t)) \geq k_0 + 1 - K(\alpha(t_f), \hat{P}(t_f)),$$

which does not necessarily minimize (10)-(11) when $k = k_0 + 1 - K(\alpha(t_f), \hat{P}(t_f))$. Formally, this idea can be expressed as

$$\begin{aligned} & m[k_0 + 1] \\ = & \mathbb{E} \left[\sum_{t=t_f}^{t_{f+1}-1} R_f(\hat{P}(t)) : \sum_{t=t_f}^{t_{f+1}-1} K(\alpha(t), \hat{P}(t)) \geq k_0 + 1 \right] \\ = & \mathbb{E} \left[R_f(\hat{P}(t_f)) + \mathbb{E} \left[\sum_{t=t_f+1}^{t_{f+1}-1} R_f(\hat{P}(t)) : \sum_{t=t_f+1}^{t_{f+1}-1} K(\alpha(t), \hat{P}(t)) \geq k_0 + 1 - K(\alpha(t_f), \hat{P}(t_f)) \middle| \mathcal{F}_{t_f} \right] \right] \\ \geq & \mathbb{E} \left[R_f(\hat{P}(t_f)) + \min_{P(t) \in \mathcal{P}} \left[\sum_{t=t_f+1}^{t_{f+1}-1} R_f(P(t)) : \sum_{t=t_f+1}^{t_{f+1}-1} K(\alpha(t), P(t)) \geq k_0 + 1 - K(\alpha(t_f), \hat{P}(t_f)) \right] \middle| \mathcal{F}_{t_f} \right] \\ = & \mathbb{E} \left[R_f(\hat{P}(t_f)) + m[k_0 + 1 - K(\alpha(t_f), \hat{P}(t_f))] \right], \end{aligned}$$

where \mathcal{F}_{t_f} denotes all the system information up until time slot t_f and the last equality follows from induction hypothesis. Finally, since $\hat{P}(t_f)$ also comes from a specific policy, it follows,

$$\begin{aligned} m[k_0 + 1] & \geq \min_{j \in \{1, \dots, |\mathcal{P}|\}} R_f(P_j) \\ & \quad + \mathbb{E} [m[k_0 + 1 - K(\alpha(t_f), P_j)]] \\ & = \min_{j \in \{1, \dots, |\mathcal{P}|\}} R_f(P_j) \\ & \quad + \sum_{i=1}^{|\mathcal{A}|} \phi(\alpha_i) m[k_0 + 1 - K(\alpha_i, P_j)] \end{aligned}$$

Overall, we proved the optimality. \square

REFERENCES

- [1] A. F. Molisch, N. B. Mehta, J. Yedidia, and J. Zhang. Performance of fountain codes in collaborative relay networks. *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4108-4119, Nov. 2007.
- [2] S. C. Draper, L. Liu, A. F. Molisch, and J. Yedidia. Routing in cooperative wireless networks with mutual-information accumulation. *Proceedings of IEEE International Conference on Communications (ICC)*, May 2008.
- [3] J. W. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE Journal on Selected Areas of Communications (JSAC)*, vol. 20, no. 8, pp. 1528-1540, Oct. 2002.
- [4] A. Shokrollahi. Raptor codes. *Proc. International Symposium on Information Theory (ISIT)*, 2004.
- [5] R. Gallager. *Discrete Stochastic Processes*. Kluwer Academic Publishers, Boston, 1996.
- [6] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
- [7] M. J. Neely. Dynamic optimization and learning for renewal systems. *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 32-46, Jan. 2013.
- [8] X. Wei and M. J. Neely. Power aware wireless file downloading: A constrained restless bandit approach. *Proc. Intl. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Hammamet, Tunisia, May 2014.
- [9] C. Li, M. J. Neely. Solving convex optimization with side constraints in a multi-class queue by adaptive $c\mu$ rule. *Queueing System*, vol. 77, no. 3, pp. 331-372, 2014.
- [10] I. Maric and R. D. Yates. Cooperative multihop broadcast for wireless networks. *IEEE Journal on Selected Areas of Communication (JSAC)*, vol. 22, no. 6, pp. 1080-1088, Aug. 2004.
- [11] M. Agarwal, J. H. Cho, L. Gao, and J. Wu. Energy efficient broadcast in wireless ad hoc networks with hitch-hiking. *Proc. IEEE INFOCOM*, March 2004.
- [12] R. Urgaonkar and M. J. Neely. Optimal Routing with Mutual Information Accumulation in Wireless Networks. *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2011.
- [13] J. Castura, Y. Mao, Rateless Coding over Fading Channels. *IEEE Communication Letters*, Vol. 10, No. 1, Jan. 2006.
- [14] J. Castura, Y. Mao, S. Draper, On Rateless Coding over Fading Channels with Delay Constraints. *IEEE Proc. ISIT*, Seattle, USA, July, 2006.
- [15] R. Garfinkel, G. Nemhauser. *Integer Programming*. John Wiley and Sons, 1972.
- [16] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol.1*. Athena Scientific, Belmont, MA.
- [17] M. J. Neely. Stability and probability 1 convergence for queueing networks via Lyapunov optimization. *Journal of Applied Mathematics*, doi:10.1155/2012/831909, 2012.
- [18] B. Fox. Markov renewal programming by linear fractional programming. *SIAM Journal on Applied Mathematics*, vol. 14, no. 6, pp. 1418-1432, Nov. 1966.