Interference Is Not Noise

Shreeshankar Bodas, Devavrat Shah
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
{bodas, devavrat}@mit.edu

Damon Wischik
Department of Computer Science
University College London
D.Wischik@cs.ucl.ac.uk

***Abstract*—This paper looks at the problem of designing wireless medium access algorithms. Inter-user interference at the receivers is an important characteristic of wireless networks. We show that decoding (or canceling) this interference results in significant improvement in the system performance over protocols that either treat interference as noise, or explicitly avoid interference at the receivers by allowing at most one of the transmitters in its range to transmit. This improvement in performance is realized by means of a medium access algorithm with: (a) polynomial computational complexity per timeslot, (b) polynomially bounded expected queue-length at the transmitters, and (c) a throughput region that is at least a poly-logarithmic fraction of the largest possible throughput-region under any algorithm operating using that treats interference as noise. Thus, the hardness of low-delay network scheduling (a result by Shah, Tse and Tsitsiklis [1]) is an artifact of explicitly avoiding interference, or treating it as noise and can be overcome by a rather simple medium access algorithm that does not require information theoretic "block codes."**

I. Introduction

We are interested in designing "efficient" medium access protocols in the context of wireless networks. In the most general case, a wireless network consists of a set of transmitters and receivers communicating over the wireless channel. The signals transmitted by the different transmitters are corrupted by noise, and also the simultaneous transmissions from different transmitters potentially interfere with each other. By "efficient" we mean a protocol that gives good throughput and delay guarantees for all the users, and has small computational complexity (polynomial in the system-size) per unit time.

A vast majority of previous research on this topic, starting with the seminal work by Tassiulas and Ephremides [2], [3] has focused on guaranteeing good throughput performance. In these network models, the so-called independent set interference model is used: a link in the communication network is represented as a node in a graph, and two nodes are connected by an edge if the transmissions on the corresponding communication links interfere with each other. The weight of a node is proportional to the packet backlog at the corresponding transmitter. The set of allowed schedules consists precisely of all the independent sets in this graph. The MaxWeight scheduler picks, in every timeslot, an independent set with the maximum weight. This algorithm is known to be throughput-optimal, but choosing "heavy" independent sets (even when all nodes have a weight = 1) is an NP-hard problem. There have been results [4], [5], [6], [7], [8], [9] for reducing the computational complexity without compromising throughput, but the expected queue-lengths at the transmitters under these schemes are known to be super-polynomial in the system-size.

These results suggest that there is a tension between the three objectives of providing high throughput, low delays, and small computational complexity. This observation was made concrete in [1] where the authors show that in the independent set interference model or the analogous SINR[1]-based decoding model, for general network graphs, there does not exist a scheduling policy that:

1. provides at least an $n^{\epsilon-1}$ fraction of the throughput region, for some $\epsilon > 0$,
2. results in $\mathcal{O}(n^K)$ expected queue-sizes at each of the transmitters, for some integer K, and
3. has $\mathcal{O}(n^M)$ computational complexity per timeslot for some integer M,

unless P=NP. We would ideally like to guarantee a good performance on each of these three metrics (throughput, delay and complexity). This brings us to the next logical question: how, if at all, can we get around this hardness result?

In this paper, we make a case for *decoding* interference, i.e., not treating it as noise, and not taking any explicit efforts to avoid it. There is a rich history of interference decoding or (more recently) interference alignment schemes in the information theory community [10], [11]. It is well-known that decoding or aligning interference improves throughput. We show that in addition, it helps in guaranteeing low[2] expected queuing delay and also low computational complexity per timeslot. Specifically, we design a medium-access algorithm (QNUB) that: (a) guarantees an arbitrarily large fraction of the maximum throughput, and (b) results in polynomially bounded expected queue-lengths and polynomially bounded number of computations per timeslot, when the load is within a poly-logarithmic fraction of the maximum throughput (which is more than the $n^{\epsilon-1}$-fraction throughput requirement for the hardness result in [1]). The main reason why the hardness result by Shah et al is circumvented is that we allow for interference between transmissions and thus, are not required to select independent sets as the only possible schedules.

The rest of this paper is organized as follows: in Section II we define the system model and the problem statement. In Section III we propose a medium access protocol (QNUB) for

[1] Signal-to-Interference and Noise Ratio

[2] In this paper, unless otherwise specified, "low" in the context of delay or complexity means "polynomially bounded in the system-size."

this system. We show that the QNUB protocol can achieve any pre-specified fraction of the throughput region for this system (Section IV, Theorem 1). In Section V we derive the conditions for a successful decoding of the symbols transmitted under QNUB. In Section VI we analyze the complexity of decoding the symbols under the QNUB protocol, and show that it is polynomially bounded in the system-size per timeslot (Theorem 3). In Section VII, we analyze the expected queue-size at the transmitters, and show that it is bounded by a constant independent of n under the QNUB protocol (Theorem 4). The last two results (constant decoding delay and constant expected queue-length) hold if the system has a load that is a poly-logarithmic fraction of the maximum possible. We conclude with a discussion of results and extensions in Section VIII. Mainly due to space limitations, a number of proofs have been omitted and presented in a technical report [12].

As a result of Theorems 1, 3 and 4, the hardness result of [1] does not hold here. While this is not terribly surprising, it is worth noting that the algorithm that we propose is distributed, performs minimal computation and does not require classical information theoretic "long block codes." We believe it (more precisely, its natural variant) would be of practical utility; it is the first step of a long term research program with an end goal of low complexity medium access that achieves high throughput and low delay for generic wireless network.

II. System Model, Problem Statement

A. System Model

We consider a system with n receivers (access points), communicating with a set of users over the wireless channel. Let the set of receivers be $\mathcal{R} = \{R_1, \ldots, R_n\}$. The system evolves in discrete time. A user in this system is associated with a *user-class*, identified by the set of receivers within its transmission range. There are $S \leq 2^n - 1$ possible user-classes. For any integer $k \geq 1$, let $[k] := \{1, \ldots, k\}$. For $\ell \in [n]$ and $s \in [S]$, let $H_{\ell,s} = 1$ if the users of class s have the receiver R_ℓ in their transmission range, and 0 otherwise.

We make the following simplifying assumption: there is exactly one user belonging to each one of the S classes. This assumption is WLOG because if there are more than one users of the same class, then we treat it as one "super-user," with the external arrivals to the individual users being replaced by one arrival process with an appropriately higher rate. Further, if we show that the expected queue-length of this "super-user" is "small," then each of the individual user-queues are "small" as well.

External arrivals: In every timeslot, a (potentially nonzero) number of files enter the system. A file is a finite collection of symbols (or bits). Let $A_s(t)$ be the indicator random variable that defines if a user-class s has a file-arrival at the end of timeslot t. We assume that the file-arrival or non-arrival in any class s is an i.i.d. Bernoulli process. That is,

$$A_s(t) = \begin{cases} 1 & \text{with probability } \lambda_s, \\ 0 & \text{with probability } 1 - \lambda_s. \end{cases}$$

We assume that the (new) file-arrivals occur just before the end of timeslot t, so the (new) arrival(s) in timeslot t, if any, are available for transmission in timeslot $t+1$, depending upon the particular medium access algorithm employed. All the random variables $\{A_s(t)\}_{s \in [S], t \in \mathbb{Z}}$ are mutually independent. When no confusion is possible, we say that a given file belongs to a class s if it arrives to the user of class s. We assume that the size (in number of symbols) of a file of class s is geometrically distributed with the parameter μ_s, independently of all other random variables. For concreteness, the reader may think of these symbols as bits or equivalently, Binary Phase-shift Keying (BPSK) symbols $\in \{-1, 1\}$, or any other constellation. In every timeslot, a receiver receives a channel-coefficient-weighted linear combination of the transmissions of the users within its range, where the channel coefficients can be random in general. For ease of exposition, we assume that the channel coefficients are all equal to 1. This is not a binding assumption for any of the results (in qualitative sense).

Departure: When each one of the symbols in the file has been successfully decoded by each one of the receivers within the corresponding user's transmission range, the file leaves the system. (The exact mechanism of decoding the symbols is protocol-dependent.)

B. Problem statement

We require successful decoding of all the symbols in every file, at each of the receivers within the transmission range of the corresponding user. Let $n_s(t)$ denote the number of files of class s present in the system at the beginning of timeslot t. Define $\rho_s := \lambda_s / \mu_s$. Our objective is to design a medium access protocol with the following properties:

1. Given any $\epsilon > 0$, it stabilizes (makes positive recurrent) the Markov chain $[n_1(t), \ldots, n_S(t)]$ as long as $\sum_{s=1}^{S} H_{\ell,s}\rho_s \leq 1/(1+2\epsilon)$ for all $\ell \in [n]$. Note that if $\rho = [\rho_s]$ can be supported by any algorithm that treats interference as noise, then these inequalities are necessarily satisfied: they are linear relaxation of the independent set constraints.
2. For some integer $K > 0$, if $\sum_{s=1}^{S} H_{\ell,s}\rho_s \leq 1/(\log n)^K$ for all $\ell \in [n]$, then the expected number of symbols (under the stationary measure) to be transmitted corresponding to the user s is polynomially bounded in n for all s, and the expected decoding complexity per timeslot is polynomially bounded in n, for each receiver R_ℓ.

Main result: We design a medium access algorithm called QNUB (Section III) that satisfies the above two requirements.

III. Medium Access Algorithm

The Medium Access (MAC) algorithm that we propose operates at the granularity of the "files," which is finer than the users. Thus effectively each user is making decision related to the MAC. However for ease of exposition, we shall provide behavioral description per-file. To that end, we define the behavior of a given file, referred to as $\star$. Consider a timeslot t, after the time in while file arrived, and has not yet departed:

it transmits a linear combination of (a subset of) its symbols. The symbols involved in the linear combination include all the symbols that were ever involved in a linear combination in any previous timeslot, and possibly one more (new) symbol. Suppose that in timeslot $t-1$, the file $\star$ transmitted a linear combination of k symbols. Then the transmission of $\star$ in timeslot t is of the form

$$y(t) = (a_1(t)x_1 + \cdots + a_k(t)x_k) + \mathbb{1}_{\text{new}} \cdot a_{k+1}(t)x_{k+1}$$

where x_i denotes the i^{th} symbol in the file, and the coefficients $a_i(t)$ are chosen to be i.i.d. $\mathcal{N}(0,1)$ random variables. Here $\mathbb{1}_{\text{new}}$ is the indicator that a "new" symbol is introduced in the previous timeslot's linear combination. It can be a function of other system parameters (not shown here explicitly). The file $\star$ stops introducing newer symbols in the linear combinations once all its symbols are already part of the linear combination, i.e., it has no more new symbols to transmit.

When a receiver, say R_ℓ is able to decode a subset of the symbols transmitted by $\star$, it sends an acknowledgment that informs $\star$ of the indices of the decoded symbols. The file $\star$ continues transmitting its linear combinations until it receives an acknowledgment from each of the intended receivers for each of its symbols, and then leaves the system.

Mathematically, define the random variables $Y_{s_j}(t)$ and $Z_{s_j}(t)$ as follows:

1. $Y_{s_j}(t) = 1$ if in timeslot t the j^{th} file of class s introduces a new symbol in the current linear combination, and 0 otherwise.
2. $Z_{s_j}(t) = 1$ if at the end of timeslot t the j^{th} file of class s has at least one more new symbol to transmit (in timeslot $t+1$ or later, as part of the linear combination), and 0 otherwise.

The random variable $Y_{s_j}(t)$ is defined by the policy, while random variable $Z_{s_j}(t)$ is defined (in general) by the file-size distribution and the transmission policy. We assume that the file-sizes for all users are geometrically distributed, so that $\mathbb{P}(Z_{s_j}(t) = 0) = \mu_s$ for all j and t. Thus, $\mathbb{E}[Z_{s_j}(t)] = 1/\mu_s - 1$, and the average size of a file of class s is $1/\mu_s$ since a file is assumed to have at least 1 symbol to begin with. Thus $\rho_s = \lambda_s/\mu_s$ denotes the average "load" of files of class s.

Let $n_s(t)$ denote the number of files of class s that are present in the system at the beginning of timeslot t. Then

$$n_s(t+1) = n_s(t) - \sum_{r=1}^{n_s(t)} Y_{s_j}(t)(1 - Z_{s_j}(t)) + A_s(t+1).$$

Each of the receivers R_ℓ maintains a price q_ℓ that is updated according to

$$q_\ell(t+1) = \left[q_\ell(t) + \alpha_\ell \left(\sum_{s=1}^{S} H_{\ell,s} \sum_{j=1}^{n_s(t)} Y_{s_j}(t) - 1 \right) \right]^+, \quad (1)$$

where α_ℓ is a small positive constant to be specified later.

There are two types of queue that naturally arise in this system, defined below.

Definition 1 (Equation-queue)**.** *For a receiver R_ℓ, define the equation queue-length $Q_\ell(t)$ at the beginning of timeslot t to be the total number of symbols that are involved in the any of the linear combinations received by R_ℓ, minus the number of equations (or linear combinations) received, until the end of timeslot $t-1$.* ◇

Definition 2 (Symbol-queue)**.** *For each user s, define the symbol queue-length $P_s(t)$ at the end of timeslot t to be the total number of symbols that are present (in one of the files) at s that have not yet been successfully decoded by at least one of the intended receivers, at the end of timeslot t.* ◇

The quantity $q_\ell(t)$ is proportional to the equation queue-length at the beginning of timeslot $t > 0$, i.e., $q_\ell(t) = \alpha_\ell Q_\ell(t)$ if $q_\ell(0) = Q_\ell(0) = 0$. At the beginning of every timeslot t for which $q_\ell(t) = 0$, the receiver R_ℓ sends (broadcasts) an acknowledgment to each one of its transmitters that all the symbols transmitted (to R_ℓ) until the beginning of timeslot t have been decoded. (That this is indeed the case is established by Theorem 2.) The files leave the system based upon these acknowledgments.

We let $Y_{s_j}(t) = 1$ with probability $x_s(t)$. This "transmission rate" $x_s(t)$ is defined as

$$x_s(t) = \min\left\{ \frac{w_s}{n_s(t) + \sum_{\ell=1}^{n} H_{\ell,s} q_\ell(t)}, 1 \right\}.$$

We choose $w_s = c\rho_s$ for some constant $c > 0$ to be chosen later. We call this scheme the Queue-length and Number of Users Based (QNUB) scheme.

Some remarks on this medium access / congestion control algorithm are in order.

(1) This medium access algorithm is motivated by [13], where the objective is to establish the connection-level stability of a class of congestion-control algorithms without the time-scale separation assumption. In [13] the authors consider a continuous-time model where each user transmits data at a "rate" $x_s(t)$, and the receivers receive a vector of the transmitted symbols (as opposed to a linear combination). This would be the case if the different users were transmitting on different carrier frequencies, and data were infinitely divisible. Our analysis of QNUB requires only minor modifications to show that even if the receivers receive a linear combination of the transmitted symbols/packets (which would be the case if all the users were transmitting on the same carrier frequency), the connection-level stability result of [13] holds, i.e., the Markov chain of the number of users ($n_s(t)$) and the prices (effective queue-lengths) at the receivers ($q_\ell(t)$) is positive recurrent if $\sum_{s=1}^{S} H_{\ell,s}\rho_s < 1$. Thus we strengthen the result of [13].

(2) The main qualitative difference between the congestion controller in [13] and QNUB is the presence of the term $n_s(t)$ in the denominator. It results in a possible loss of throughput: depending upon the choice of c, we can only establish the positive recurrence of the Markov chain $[q_\ell(t), n_s(t)]_{\ell,s}$ for a constant fraction of the throughput region. However, this modification helps in keeping the expected decoding delay within a polynomial of the system-size (n).

(3) Another difference between the two algorithms is that each user needs to know $n_s(t)$ and ρ_s. A simple approach would require communication between the receivers to calculate (or estimate) these quantities. We plan to address this issue in the future work.

IV. Throughput Performance

We now analyze the throughput performance of the QNUB algorithm. The following two technical lemmas are useful in establishing the desired result (Theorem 1). Their proofs are straightforward, and available in the tech report [12].

Lemma 1. $\mathbb{E}[n_s(t+1)^2 - n_s(t)^2 \mid n_s(t), x_s(t)] = 2n_s(t)(\lambda_s - n_s(t)x_s(t)\mu_s) + (\lambda_s - n_s(t)x_s(t)\mu_s)^2 + \lambda_s(1-\lambda_s) + n_s(t)x_s(t)\mu_s(1 - x_s(t)\mu_s)$.

Proof: Please see Appendix A. ∎

Lemma 2. *Define* $\alpha_{\max} := \max_{1\le i\le n} \alpha_i$, $L_0 := \max_{1\le \ell\le n} \sum_{s=1}^{S} H_{\ell,s}$ *and* $S_0 := \max_{1\le s\le S} \sum_{\ell=1}^{n} H_{\ell,s}$. *Then*

$$\mathbb{E}\left[\sum_{\ell=1}^{n} \frac{q_\ell^2(t+1) - q_\ell^2(t)}{2\alpha_\ell} \middle| \underline{n}(t), \underline{q}(t)\right] \le -\sum_{\ell=1}^{n} q_\ell(t) + n\alpha_{\max}$$
$$+ \sum_{s=1}^{S}\left\{\sum_{\ell=1}^{n} q_\ell(t) H_{\ell,s}\right\} n_s(t)x_s(t)$$
$$+ \alpha_{\max} L_0 S_0 \sum_{s=1}^{S}\left(n_s^2(t)x_s^2(t) + n_s(t)x_s(t)(1 - x_s(t))\right).$$

Proof: Please see [12]. ∎

Theorem 1. *Fix any* $\epsilon > 0$. *If* $(1+2\epsilon)\sum_{s=1}^{S} H_{\ell,s}\rho_s < 1$ *for all* ℓ, *then choosing* $c > (1+\epsilon)^2/\epsilon$ *and* $\alpha_\ell \in \left(0, \frac{\mu_s w_s}{100 L_0 S_0 (1+\epsilon)\rho_s}\right)$ *for all* $\ell \in [n]$ *in the QNUB algorithm makes the Markov chain* $[\underline{n}(t), \underline{q}(t)]$ *is positive recurrent.*

Proof: This proof is based on the same ideas as the stability proof in [13]. Please see [12] for a detailed proof. ∎

Thus the QNUB algorithm achieves any pre-specified fraction of the largest possible throughput region under any algorithm, $\mathcal{X} := \{\rho = [\rho_1, \ldots, \rho_s] : \sum_{s=1}^{S} H_{\ell,s}\rho_s < 1 \ \forall\ \ell \in [n]\}$.

V. Decodability

Our objective now is to establish that for the receiver R_ℓ, at the end of every timeslot t such that $q_\ell(t) = 0$, all the symbols that it has received so far (until the end of timeslot t) are decodable (Theorem 2).

Definition 3 (Permutation-type matrix)**.** *A matrix* $A \in \{0,1\}^{n\times n}$ *is said to be a permutation-type matrix if there exists a permutation* σ *of* $\{1, 2, \ldots, n\}$ *such that* $A_{i,\sigma(i)} = 1$ *for all* $1 \le i \le n$.

Lemma 3. *Consider a matrix* $M \in \Re^{n\times n}$ *with* $M_{ij} \sim \mathcal{N}(0,1)$ *for all* $1 \le i, j \le n$. *Let* $A \in \{0,1\}^{n\times n}$ *be a permutation-type matrix. Let* $B \in \Re^{n\times n}$ *be obtained by the element-wise product of* M *and* A, *i.e.,* $B_{ij} = M_{ij}A_{ij}$. *Then* $\mathrm{rank}(B) = n$ *with probability* 1.

Proof: Please see [12]. ∎

Remark 1. *The result clearly holds for any continuous distribution on* $\Re$.

Theorem 2. *Fix any* $\ell \in [n]$. *Let* $q_\ell(0) = 0$ *and let* $T > 0$ *be the smallest integer (time-index) such that* $q_\ell(T) = 0$. *Then with probability* 1, *all the symbols that are transmitted in any of the timeslots in* $\{0, \ldots, T-1\}$, *intended to be received by the receiver* R_ℓ, *are decoded by the receiver* R_ℓ *at the end of timeslot* $T-1$.

Proof: Please see [12]. ∎

The main idea behind the proof of Theorem 2 is that if $q_\ell(0) = q_\ell(t) = 0$, then from the beginning of timeslot 0 until the beginning of timeslot t, the receiver R_ℓ has received t linear combinations of t symbols. Solving this linear system involves inverting a matrix of coefficients, which is possible because the coefficients are chosen to be i.i.d. $\mathcal{N}(0,1)$ (or any other continuous distribution).

VI. Decoding Delay Analysis

Our objective here is to analyze the return time to 0 of the equation-queue Q_ℓ (corresponding to the receiver R_ℓ). The return time is denoted by T_ℓ, and defined as

$$T_\ell = \min\{k > 0 : Q_\ell(k) = 0, Q_\ell(0) = 0\}.$$

The reason we are interested in the return time is that from Theorem 2, a receiver R_ℓ can decode all the symbols received in the timeslots $\{0, \ldots, T_\ell - 1\}$ at the end of the timeslot T_ℓ by inverting a matrix of dimensions $T_\ell \times T_\ell$, and the complexity of this operation is $\mathcal{O}(T_\ell^3)$ computations over T_ℓ timeslots (there are faster algorithms, but the simple matrix inversion algorithm suffices our purpose). Hence if T_ℓ is "large," say $T_\ell \sim e^n$, then the matrix inversion is a time-consuming step, potentially hurting the "low complexity" part of the overall scheme. The main result here is Theorem 3 which shows that all the moments (in particular, the third moment) of the return time to 0 of the queue Q_ℓ are bounded independently of n, if the load is sufficiently small.

From Theorem 1, given $\epsilon > 0$, with the choice $c > (1+\epsilon)^2/\epsilon$, any load vector $[\rho_1, \ldots, \rho_s]$ satisfying $\sum_{s=1}^{s} H_{\ell,s}\rho_s \le 1/(1+2\epsilon)$ for all ℓ is stabilized by the QNUB algorithm. Consider the choice $c = (\log n)^4$. For n large enough, we have $c > (1+\epsilon)^2/\epsilon$ for any pre-specified ϵ, implying at least a fraction $1/3$ (corresponding to $\epsilon = 1$) of the throughput region for n large enough (in fact, for $n \ge 5$). Consider a vector $\rho = [\rho_1, \ldots, \rho_S]$ such that $\sum_{s=1}^{S} H_{\ell,s}\rho_s \le 1/(\log n)^5$. Since the entire throughput region is given by $\mathcal{X} = \{[\rho_1, \ldots, \rho_s] : \sum_{s=1}^{s} H_{\ell,s}\rho_s < 1\}$, we have $\rho \in (1/(\log n)^5)\mathcal{X}$. Our objective is to show that for this choice of ρ, the equation queue-length has a "small" expected size, as follows.

Definition 4 (Stochastic dominance)**.** *Given random variables* X *and* Y, *we say that* X *is stochastically dominated by* Y *and write* $X \le_{st} Y$ *if for all* $z \in \Re, \mathbb{P}(X \le z) \ge \mathbb{P}(Y \le z)$. ◇

Equation queue-length evolution: Choose n large enough (say $n \ge 15$) such that $2/\log n \le 3/4$. The number of arrivals to the equation-queue Q_ℓ at the beginning of timeslot t is distributed according to

$$
\begin{aligned}
J_\ell(t) &\overset{d}{=} \sum_{s:H_{\ell,s}=1} B(n_s(t), x_s(t)) \\
&\leq_{st} \sum_{s:H_{\ell,s}=1} B\left(n_s(t), \frac{c\rho_s}{n_s(t)}\right) \\
&\leq_{st} \sum_{s:H_{\ell,s}=1} Poi(2c\rho_s) \\
&\overset{d}{=} Poi\left(2c\sum_{s=1}^{S} H_{\ell,s}\rho_s\right) \leq_{st} Poi\left(\frac{2}{\log n}\right), \qquad (2)
\end{aligned}
$$

where the summation notation is used to describe a sum of independent random variables with the specified distribution. Here the first inequality holds because:

1) if $X \sim B(n,p)$ and $Y \sim B(n,q)$ with $p < q$, then $X \leq_{st} Y$ (see [12]), and
2) if $X \leq_{st} X_1$ and $Y \leq_{st} Y_1$, and if the random variables X, X_1, Y, Y_1 are mutually independent, then $X + Y \leq_{st} X_1 + Y_1$ (see [12]).

The second inequality holds because:

1) $B(n, \delta/n) \leq_{st} Poi(2\delta)$ as long as $1 - \delta \geq e^{-2\delta}$, which holds if $\delta \leq 0.75$ (see [12]), and
2) if $X \leq_{st} X_1$ and $Y \leq_{st} Y_1$, and if the random variables X, X_1, Y, Y_1 are mutually independent, then $X + Y \leq_{st} X_1 + Y_1$ (see [12]).

The second-last equality holds because the sum of independent Poisson random variables is a Poisson random variable, while the last stochastic dominance holds because if $0 < \alpha < \beta$ and $X \sim Poi(\alpha)$ and $Y \sim Poi(\beta)$, then $X \leq_{st} Y$.

In reference to Equation (2), our objective now is to show that the length of Q_ℓ at the beginning of timeslot t, i.e., $Q_\ell(t)$, is stochastically dominated by the length of a single-server queue, served by a deterministic, unit-capacity server, and having i.i.d. $Poi(2/\log n)$ arrivals in each timeslot. This construction is useful because it is easier to handle a queuing system with i.i.d. arrivals.

Lemma 4. *Consider two queues Q_ℓ and R, each served by a single server with a deterministic capacity of 1 packet per timeslot. The evolution of Q_ℓ is given by*

$$Q_\ell(t+1) = (Q_\ell(t) + J_\ell(t) - 1)^+ ,$$

where $J_\ell(t)$ is given by Equation (2). The evolution of $R(t)$ is given by

$$R(t+1) = (R(t) + B(t) - 1)^+ ,$$

where $B(t) \sim Poi(2/\log n)$, i.i.d., independent of all other random variables, and with $\log n \geq 8/3$. Let $Q_\ell(0) = R(0) = 0$. Then $Q_\ell(t) \leq_{st} R(t)$ for all $t \geq 0$.

Proof: The proof follows an inductive argument. Please see [12] for details. ∎

We now analyze the return time to 0 of a Markov chain with i.i.d. $Poi(\theta)$ arrivals and deterministic service of 1 packet per timeslot, with $\theta < 1$ for stability. We are interested in the case when θ is small, and assume that $\theta < 1/(2e)$. The main conclusion of Theorem 3 is that each of the moments of the return time to 0 of the queue under consideration is finite, and bounded by a constant independent of the system-size n. Thus, the expected number of operations required for the matrix-inversion operation mentioned at the beginning of this section is finite, independent of n as long as $2/\log n < 1/(2e)$ or $\log n > 4e$.

Theorem 3. *Consider two queues Q_ℓ and R, each served by a single server with a deterministic capacity of 1 packet per timeslot. The evolution of Q_ℓ is given by*

$$Q_\ell(t+1) = (Q_\ell(t) + J_\ell(t) - 1)^+ ,$$

where $J_\ell(t)$ is given by Equation (2). The evolution of $R(t)$ is given by

$$R(t+1) = (R(t) + B(t) - 1)^+ ,$$

where $B(t) \sim Poi(2/\log n)$, i.i.d., independent of all other random variables, and with $\log n \geq 8/3$. Let $Q_\ell(0) = R(0) = 0$. Define $T_\ell := \min\{t \geq 1 : Q_\ell(t) = 0\}$. Then for $\theta < 1/(2e)$, we have $\mathbb{P}(T_\ell \geq k) \leq 2(e\theta)^k$ for all $k \geq 1$. Further, for all $m \geq 1$,

$$\mathbb{E}[T_\ell^m] \leq 2\left(\left(\frac{m}{\log 2} + 1\right)^{m+1} + m! \cdot \left(\frac{1}{\log 2}\right)^{m+1}\right).$$

Consequently, the expected decoding complexity (T_ℓ^3 computations over T_ℓ timeslots, or $\mathbb{E}[T_\ell^2]$ computations per timeslot) at each of the receivers R_ℓ is constant per timeslot.

Proof: Please see Appendix C. ∎

As a consequence of Theorem 3, we show that the expected value of the maximum of n return-time random variables $T_1, \ldots, T_n$ is $\mathcal{O}(\log n)$. This claim does not require the queues to have independent arrival processes as long as each arrival process is uniformly dominated by an i.i.d. Poisson arrival process for all timeslots.

Lemma 5. *Consider a system $[Q_1, \ldots, Q_n]$ of n single-server queues. Let $T = \max(T_1, \ldots, T_n)$, where T_i is the return time to 0 of Q_i. Let the number of arrivals to Q_i at the beginning of timeslot t be $J_i(t)$. Let $J_i(t) \leq_{st} Poi(\theta)$ for all i and t, with $2e\theta < 1$. Each of the queues Q_i is served a constant service rate of 1 packet per timeslot. Then there exists $a_0 > 0$ such that $\mathbb{E}[T] \leq a_0 \log n$ for all n large enough.*

Proof: Please see [12]. ∎

VII. Expected Queue-length Analysis

Let P_s denote the symbol-queue maintained by the user corresponding to the class s, and $P_s(t)$ denote the queue-length at the end of timeslot t. (See Definition 2 for the definition of $P_s(t)$.) The main result here (Theorem 4) is that if the system has a sufficiently light load, then the expected value of $P_s(t)$ is bounded by a constant independent of n.

The arrival process for P_s: If $A'_s(t)$ denotes the number of arrivals to P_s at the beginning of timeslot t, then

$$A'_s(t) = \begin{cases} 0 & \text{with probability } 1 - \lambda_s \\ k & \text{with probability } \lambda_s(1-\mu_s)^{k-1}\mu_s, \qquad k \geq 1. \end{cases}$$

The departure process for P_s: A given symbol leaves the symbol-queue P_s when it has been successfully decoded by each one of the intended receivers (i.e., every receiver within the transmission range of the user of class s). For ease of analysis, we consider a somewhat wasteful medium access strategy as defined below, which we continue to call QNUB.

1) Starting with $Q_\ell(0) = 0$ for all ℓ, let the medium access algorithm QNUB run on its own and the queues evolve accordingly.
2) When a given queue, say Q_ℓ becomes 0, it informs its transmitters by broadcasting a message. The transmitters continue transmitting according to the policy (QNUB), but note the last new symbol that was transmitted before receiving the acknowledgment that Q_ℓ has become 0. Each transmitter notes the first time it hears from a receiver that its queue has become 0, and calls that timeslot a "marker."
3) When each of the queues has become 0 at least once, say at the end of timeslot T_0, each of the transmitters (with the understanding that each one of the symbols transmitted before the marker has been decoded by each one of the intended receivers) start transmitting the (linear combinations of the) symbols that were transmitted after the marker. Each of the receivers starts with empty queues (ignoring all the linear equations received in the time when its queue first became empty, until the current timeslot T_0), and the process continues. $\diamond$

This strategy is wasteful as compared to the original QNUB strategy because the receivers ignore all the linear combinations received after the first time-instant when their equation-queues become zero, until that time when each of the equation-queues becomes 0 at least once, and the process repeats. The advantage is that this strategy is easier to analyze, and yields bounds on the expected symbol queue-lengths of the original system. (It is possible to formally show this dominance, using standard arguments similar to those in the proof of Lemma 4.)

We first consider a system with the arrival process as described by $A'_s(t)$, and deterministic service of 1 packet (or symbol) per timeslot. We show that as long as $\rho_s = \lambda_s/\mu_s < 1$, the system is stable and has an exponential decay of the probability measure.

Lemma 6. *Consider a single-server queue with deterministic service of* 1 *packet per timeslot, and the number of arrivals in every timeslot is equal to* $A'_s(t)$*, i.i.d. across timeslots, with* $\lambda_s/\mu_s < 1$*. Then the stationary distribution of the resultant queue-length Markov chain is given by*

$$\pi_k = \pi_0 \lambda_s \left(\frac{1-\mu_s}{1-\lambda_s} \right)^k, \quad k \geq 1$$

with $\pi_0 = (\mu_s - \lambda_s)/(\mu_s(1-\lambda_s))$.

Proof: Elementary flow-balance. ■

An immediate consequence of Lemma 6 is that the expected queue-length in a queuing system with $A'_s(t)$ arrivals and 1 unit of service in every timeslot is constant. Now the actual system at hand is such that it has $A'_s(t)$ arrivals in every timeslot, with (presumably) $\rho_s \ll 1$, but the service process is governed by a complex protocol. We show that as long as the load is a poly-logarithmic fraction of the maximum possible load, even this system has a "small" expected queue-size. We need to consider a lightly loaded system because the exact system is difficult to analyze. This restriction is acceptable towards our final goal, because the hardness result in [1] applies as long as the throughput achieved under the candidate protocol is at least an $n^{\epsilon-1}$ fraction of the maximum possible for some $\epsilon > 0$. We believe that the result can be proved without the poly-logarithmic factors, but do not consider it for this paper.

Theorem 4. *Consider the choice* $c = (\log n)^4$ *in the QNUB algorithm. Let the load be such that* $\sum_{s=1}^{S} H_{\ell,s}\rho_s \leq 1/(\log n)^5$*. Then for* n *large enough, the expected length of* P_s *under the stationary distribution is bounded by* $\frac{8a_0 r_0}{3\mu_s}$ *for some constant* r_0*, and* a_0 *as defined in the statement of Lemma 5.*

Proof: Please see [12]. ■

Thus, we have established that the expected queue-length at each of the transmitters is polynomially bounded in the system-size n (in fact, bounded by a constant) as long as the load is within a poly-logarithmic fraction of the maximum possible load.

VIII. Conclusions and Discussion

We considered a wireless network with a fixed set of n receivers and a fixed set of transmitters. Each users sees external file-arrivals. A file consists of a finite number of symbols that need to be transmitted to each of the receivers in the transmission range. For this system, we developed a medium access algorithm (QNUB) that achieves any pre-specified fraction of the throughput region. Further, if the load is within a poly-logarithmic fraction of the maximum, then the expected queue-length at each transmitter and the decoding complexity per timeslot are constant independent of the system-size (n). Thus, decoding interference not only helps improve throughput, but also reduce computational complexity and yields a better delay performance, in contrast with the result by Shah et al [1] that in a general wireless network with the independent set scheduling constraint, these three objectives are unattainable unless P=NP.

This result can be strengthened to account for the following modifications to the system-model.

(1) **Noise:** Consider a system where additive white Gaussian noise corrupts the received symbols at the receiver R_ℓ. In this case the received signal can be written as $y = Vx + w$. A possible transmission strategy is transmitting $M - PAM$ symbols for x, and matrix-inversion and quantization-based decoding: $(V^TV)^{-1}V^Ty = x + (V^TV)^{-1}V^Tw$. Here V denotes the coefficient matrix, with each entry $V_{ij} \sim \mathcal{N}(0,1)$, i.i.d. This scheme can be implemented by means of matrix inversion and component-by-component quantization to an $M - PAM$ lattice, and has complexity that is polynomial in the size of V. This scheme would result in a finite error probability (owing to noise), but error-correction at higher

layers (resulting in retransmissions) can be used to achieve a poly-logarithmic (constant?) fraction of the maximum possible throughput of $\frac{1}{2}\log(1+SNR)$.

(2) **Poly-logarithmic fractions:** Theorems 3 and 4 show that if the system has a load that is a poly-logarithmic fraction of the maximum possible load, the expected decoding complexity and the expected queue-size at each transmitter are polynomially bounded (in fact, constant) in the system-size. We believe that the poly-logarithmic fractions are artifacts of our proof techniques: because the QNUB protocol is difficult to analyze in closed-form, we need an appropriate concentration of measure for the arrival process to derive analytical results. As Lemma 6 shows, if the service process is "simple," then the expected queue-size can be proved to be constant, independent of the system-size n at any fixed fraction of the maximum possible load.

(3) **Distributed implementation:** The proposed QNUB algorithm is not entirely distributed, because each transmitter needs to know $n_s(t)$ and ρ_s. In the specific system we considered (1 transmitter for each class, with an appropriately higher external arrival rate), the transmitters know $n_s(t)$ and can (locally) estimate ρ_s. But when more than 1 users of the same class are present, we need communication between the receivers (among other possibilities) to implement the protocol. Modifying the protocol to replace $n_s(t)$ with each transmitter's own symbol queue-length is one possible solution. (The term ρ_s gets automatically adjusted, because if we have two users U_1 and U_2 of the same class, with external file arrival rates λ and η (instead of 1 user U of the class with an external file arrival rate of $\lambda+\eta$), then they can each (locally) estimate λ/μ_s and η/μ_s and transmit with appropriately smaller attempt probabilities.) In the future work, we plan to study this aspect of the problem.

ACKNOWLEDGMENTS

This work has been supported in part by the DARPA ITMANET project.

REFERENCES

[1] D. Shah, D. N. C. Tse, and J. Tsitsiklis, "Hardness of low delay network scheduling," *IEEE Trans. Inform. Theory*, vol. to appear, 2011.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 4, pp. 1936–1948, December 1992.

[3] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inform. Theory*, vol. 39, pp. 466–478, March 1993.

[4] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proc. IEEE Infocom*, Mar. 1998.

[5] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. Ann. ACM SIGMETRICS Conf.*, 2006.

[6] L. Jiang, D. Shah, J. Shin, and J. Walrand, "Distributed random access algorithm: Scheduling and congestion control," *IEEE Trans. Inform. Theory*, vol. 56, no. 12, pp. 6182–6207, Dec. 2010.

[7] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Network.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.

[8] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *Proc. Ann. ACM SIGMETRICS Conf.*, Jun. 2009.

[9] J. Shin and D. Shah, "Randomized scheduling algorithm for queuing networks," *The Annals of Applied Probability*, vol. to appear, 2011.

[10] D. N. C. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[11] V. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[12] S. Bodas, D. Shah, and D. Wischik, "Interference is not noise: Tech report," http://www.mit.edu/~bodas/intf_noise_tech_report.pdf, 2011.

[13] X. Lin, N. Shroff, and R. Srikant, "On the connection-level stability of congestion-controlled communication networks," *IEEE Trans. Inform. Theory*, vol. 54, no. 5, pp. 2317–2338, May 2008.

[14] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge: Cambridge University Press, 2005.

APPENDIX A
PROOF OF LEMMA 1

Let $\Delta = \Delta_s(t) := A_s(t+1) - \sum_{j=1}^{n_s(t)} Y_{s_j}(t)(1 - Z_{s_j}(t))$. Then $n_s(t+1) = n_s(t) + \Delta$, and $\mathbb{E}[n_s(t+1)^2 - n_s(t)^2 \mid n_s(t), x_s(t)] = 2n_s(t)\mathbb{E}[\Delta \mid n_s(t), x_s(t)] + \mathbb{E}[\Delta^2 \mid n_s(t), x_s(t)]$.

We have $\mathbb{E}[\Delta \mid n_s(t), x_s(t)] = \lambda_s - n_s(t)x_s(t)\mu_s$ and, by the independence of $A_s(t+1)$ and $\sum_{j=1}^{n_s(t)} Y_{s_j}(t)(1-Z_{s_j}(t))$,

$$\begin{aligned}\mathbb{E}[\Delta^2 \mid n_s(t), x_s(t)] &= (\lambda_s - n_s(t)x_s(t)\mu_s)^2 + \lambda_s(1-\lambda_s) \\ &\quad + n_s(t)x_s(t)\mu_s(1 - x_s(t)\mu_s).\end{aligned}$$

The result follows by an addition of the above terms.

APPENDIX B
PROOF OF LEMMA 2

From Equation (1), we have

$$\begin{aligned}&\mathbb{E}\left[\sum_{\ell=1}^{n} \frac{q_\ell^2(t+1) - q_\ell^2(t)}{2\alpha_\ell} \middle| \underline{n}(t), \underline{q}(t)\right] \\ &\leq \mathbb{E}\left[\sum_{\ell=1}^{n} q_\ell(t)\left\{\sum_{s=1}^{S} H_{\ell,s}\sum_{j=1}^{n_s(t)} Y_{s_j}(t) - 1\right\}\right. \\ &\quad \left. + \sum_{\ell=1}^{n}\frac{\alpha_\ell}{2}\left\{\sum_{s=1}^{S} H_{\ell,s}\sum_{j=1}^{n_s(t)} Y_{s_j}(t) - 1\right\}^2 \middle| \underline{n}(t), \underline{q}(t)\right]. \quad (3)\end{aligned}$$

We have

$$\begin{aligned}T_1 &= \sum_{\ell=1}^{n} q_\ell(t)\left\{\sum_{s=1}^{S} H_{\ell,s}\sum_{j=1}^{n_s(t)} Y_{s_j}(t) - 1\right\} \\ &= \sum_{s=1}^{S}\left\{\sum_{\ell=1}^{n} q_\ell(t)H_{\ell,s}\right\}\left(\sum_{j=1}^{n_s(t)} Y_{s_j}(t)\right) - \sum_{\ell=1}^{n} q_\ell(t),\end{aligned}$$

implying

$$\begin{aligned}&\mathbb{E}[T_1 \mid \underline{n}(t), \underline{q}(t)] \\ &= \sum_{s=1}^{S}\left\{\sum_{\ell=1}^{n} q_\ell(t)H_{\ell,s}\right\} n_s(t)x_s(t) - \sum_{\ell=1}^{n} q_\ell(t).\end{aligned}$$

Further,

$$\begin{aligned}
T_2 &= \sum_{\ell=1}^{n} \frac{\alpha_\ell}{2}\left[\sum_{s=1}^{S} H_{\ell,s}\sum_{j=1}^{n_s(t)} Y_{s_j}(t) - 1\right]^2 \\
&\le \sum_{\ell=1}^{n} \alpha_\ell \left(\left[\sum_{s=1}^{S} H_{\ell,s}\sum_{j=1}^{n_s(t)} Y_{s_j}(t)\right]^2 + 1\right) \\
&\le \sum_{\ell=1}^{n} \alpha_\ell \underbrace{\left(\sum_{s=1}^{S} H_{\ell,s}\right)}_{\le L_0} \left(\sum_{s=1}^{S} H_{\ell,s}\left(\sum_{j=1}^{n_s(t)} Y_{s_j}(t)\right)^2\right) \\
&\quad + n\alpha_{\max} \\
&\le L_0 \sum_{s=1}^{S} \underbrace{\left(\sum_{\ell=1}^{n} \alpha_\ell H_{\ell,s}\right)}_{\le \alpha_{\max} S_0} \left(\sum_{j=1}^{n_s(t)} Y_{s_j}(t)\right)^2 + n\alpha_{\max} \\
&\le \alpha_{\max} L_0 S_0 \sum_{s=1}^{S}\left(\sum_{j=1}^{n_s(t)} Y_{s_j}(t)\right)^2 + n\alpha_{\max}.
\end{aligned}$$

Conditioned on $[\underline{n}(t), \underline{q}(t)]$, the random variable $\sum_{j=1}^{n_s(t)} Y_{s_j}(t)$ is a binomial random variable $B(n_s(t), x_s(t))$, and the expected value of its square is $n_s^2(t)x_s^2(t) + n_s(t)x_s(t)(1 - x_s(t))$, and the result follows.

Appendix C
Proof of Theorem 3

Define $S := \min\{t \ge 1 : R(t) = 0\}$. Fix any integer $k \ge 0$. Generate a sequence of arrivals $[J_\ell(1), \dots, J_\ell(k)]$ and $[B(1), \dots, B(k)]$ according to the appropriate joint distributions (with B_i being i.i.d.) and $J_\ell(t) \le B(t)$ for all $1 \le t \le k$. We first show that such a construction is possible.

Consider a pair of random variables $[X, Y]$ and another pair $[P, Q]$ such that $X \le_{st} P$ and $Y \le_{st} Q$, with continuous and strictly increasing CDFs (other special cases can be handled similarly), denoted by $F_{XY}(x, y)$ and $F_{PQ}(x, y)$. Suppose further that $F_{XY}(x, y) \ge F_{PQ}(x, y)$ for all $(x, y) \in \Re^2$. Our objective is to define random variables X_1, Y_1, P_1, Q_1 such that $X_1 \le P_1$ a.s., $Y_1 \le Q_1$ a.s., $[X, Y] \overset{d}{=} [X_1, Y_1]$ and $[P, Q] \overset{d}{=} [P_1, Q_1]$. Consider the probability space $(\Omega, \mathcal{F}, \mathbb{P}) = ((0, 1)^2, \mathcal{B}((0, 1)^2), \lambda)$ where $\mathcal{B}$ denotes the Borel σ-algebra and λ, the Lebesgue measure. For any point $(x, y) \in (0, 1)^2$ define $[X_1, Y_1](x, y) = F_{XY}^{-1}(x, y)$, and $[P_1, Q_1](x, y) = F_{PQ}^{-1}(x, y)$. We therefore have $X_1 \le P_1$ a.s., $Y_1 \le Q_1$ a.s., and $[X, Y] \overset{d}{=} [X_1, Y_1]$ and $[P, Q] \overset{d}{=} [P_1, Q_1]$, as desired.

Now for the queuing system under consideration, let $k = 2$ (the cases $k = 0, 1$ are trivial and standard respectively). For any pair $(x, y) \in \Re^2$ we have

$$\begin{aligned}
&\mathbb{P}(J_\ell(1) \le x, J_\ell(2) \le y) \\
&\quad = \mathbb{P}(J_\ell(1) = x)\mathbb{P}(J_\ell(2) \le y \mid J_\ell(1) \le x) \\
&\quad \overset{(a)}{\ge} \mathbb{P}(B(1) \le x)\mathbb{P}(J_\ell(2) \le y \mid J_\ell(1) \le x) \\
&\quad \overset{(b)}{\ge} \mathbb{P}(B(1) \le x)\mathbb{P}(B(2) \le y) \\
&\quad = \mathbb{P}(B(1) \le x, B(2) \le y).
\end{aligned}$$

Here the inequality (a) holds because $J_\ell(1) \le_{st} B(1)$ and the inequality (b) holds because the random variable $B(2)$ *uniformly* stochastically dominates each one of the possible conditional distributions of $J_\ell(2)$. Hence the desired construction is possible for the case $k = 2$ and analogously, for any integer $k \ge 0$.

Now let $E_k = \{T_\ell > k\}$ denote the event that the return time to 0 of Q_ℓ is greater than $k \ge 0$. Let $G_k = \{S > k\}$. Since (on the equivalent probability space) $Q_\ell(t) \le R(t)$ for $1 \le t \le k$, we have $E_k \Rightarrow G_k$ and $\mathbb{P}(E_k) \le \mathbb{P}(G_k)$. Let F_k denote the event that in the k timeslots $\{1, \dots, k\}$, the system R has a total of at least $k + 1$ arrivals. Since the service is deterministic at 1 packet per timeslot, we have $G_k \Rightarrow F_k$ and $\mathbb{P}(G_k) \le \mathbb{P}(F_k)$.

The number of arrivals in k timeslots is a Poisson random variable with parameter $k\theta$. From Lemma 7.3 in [14], $m! \ge \sqrt{2\pi m}(m/e)^m$ for all $m \ge 0$. Hence,

$$\begin{aligned}
\mathbb{P}(F_k) &= \sum_{r=k+1}^{\infty} \frac{e^{-k\theta}(k\theta)^r}{r!} \le \sum_{r=k+1}^{\infty} \frac{e^{-k\theta}k^r\theta^r e^r}{\sqrt{2\pi r}\cdot r^r} \\
&\le \sum_{r=k+1}^{\infty} (e\theta)^r \le 2(e\theta)^{k+1},
\end{aligned}$$

since $e\theta < 1/2$. Therefore, for all $k \ge 1, \mathbb{P}(E_{k-1}) = \mathbb{P}(T_\ell \ge k) \le 2(e\theta)^k$.

Now $\mathbb{E}[T_\ell^m] = \sum_{r=1}^{\infty} r^m \mathbb{P}(T_\ell = r) \le \sum_{r=1}^{\infty} r^m \mathbb{P}(T_\ell \ge r) \le 2\sum_{r=1}^{\infty} r^m (e\theta)^r$. To bound $\sum_{r=1}^{\infty} r^m (e\theta)^r$, let $p := e\theta$. Note that the function $f(x) = x^m p^x$ reaches its maximum at $x^* = m/\log(1/p)$, and monotonically decreases after x^*. Hence

$$\begin{aligned}
\sum_{r=\lceil x^*\rceil+1}^{\infty} r^m (e\theta)^r &\le \int_{\lceil x^*\rceil}^{\infty} x^m (e\theta)^m dx \\
&\le \int_0^{\infty} x^m (e\theta)^x dx \\
&= m! \cdot \left(\frac{1}{\log(1/(e\theta))}\right)^{m+1}.
\end{aligned}$$

Further,

$$\sum_{r=1}^{\lceil x^*\rceil} r^m p^r \le \lceil x^*\rceil \cdot \max((\lceil x^*\rceil)^m p^{\lceil x^*\rceil}, (\lfloor x^*\rfloor)^m p^{\lfloor x^*\rfloor}) \le \lceil x^*\rceil^{m+1}.$$

Combining the above inequalities and noting that $\lceil x^*\rceil \le m/\log(1/p) + 1$, we get

$$\sum_{r=1}^{\infty} r^m (e\theta)^r \le \left(\frac{m}{\log(1/(e\theta))} + 1\right)^{m+1} + m! \cdot \left(\frac{1}{\log(1/(e\theta))}\right)^{m+1}.$$

Since $e\theta < 1/2$, this translates to a looser upper-bound,

$$\mathbb{E}[T_\ell^m] \le 2\left(\left(\frac{m}{\log 2} + 1\right)^{m+1} + m! \cdot \left(\frac{1}{\log 2}\right)^{m+1}\right),$$

as desired.