Integration of Streaming and Elastic Traffic in Wireless Networks

Sem Borst*†, Nidhi Hegde‡

*Alcatel-Lucent, Bell Laboratories, P.O. Box 636, Murray Hill, NJ 07974-0636, USA

†Department of Mathematics & Computer Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

‡France Telecom R&D, 38-40 rue du Général Leclerc, 92794 Issy-les-Moulineaux, France

***Abstract*— Channel-aware scheduling strategies have emerged as an effective mechanism for improving the throughput of wireless data users by exploiting rate variations. The improvement in throughput comes however at the expense of an increase in the variability of the service rate received over time. While the larger variability only has a limited impact on delay-tolerant data transfers, it does severely affect delay-sensitive applications. In order to examine the merits of channel-aware scheduling for the latter users, we consider a wireless system supporting a combination of streaming and elastic traffic. We first examine a scenario with rate-adaptive streaming traffic, and analyze the flow-level performance in terms of transfer delays and user throughputs for various canonical resource sharing schemes. Simulation experiments demonstrate that the analytical results yield remarkably accurate estimates, and indicate that channel-aware scheduling achieves significant performance gains. Next we investigate a scenario where the streaming sources have an intrinsic rate profile and stringent delay requirements. In that case, channel-aware scheduling yields only modest performance gains, and may even be harmful.**

I. Introduction

Wireless networks are rapidly evolving to support a wide variety of high-speed data applications, in addition to conventional voice services and current low-bandwidth data services such as short messaging. The integration of these heterogeneous applications on a common platform raises similar challenges as in wireline integrated networks. In wireless environments, these issues are further exacerbated by interference problems, intrinsically limited bandwidth, and highly variable and unpredictable propagation characteristics. Specifically, the channel quality may dramatically differ among spatially distributed users due to distance-related attenuation. In addition, the channel conditions for a given user may significantly vary over time because of fading effects.

Wireless voice networks typically rely on power control mechanisms for adjusting the transmit power to compensate for the varying channel quality and maintain a fixed transmission rate. Various data applications on the other hand, such as file transfers and Web browsing sessions, do not have a stringent rate requirement and are less sensitive to packet-level delays. Such *elastic* applications are well-suited for rate control algorithms which dynamically adapt the transmission rate over time so as to track the fluctuations in channel quality while transmitting at constant power.

The capability of dynamic rate control, combined with the relative delay tolerance of data applications, opens up the possibility of scheduling transmissions so as to achieve throughput gains. A particularly attractive approach is to use *channel-aware* scheduling strategies, such as the Proportional Fair algorithm for the CDMA 1xEV-DO system [1]–[3], which schedule the transmissions to the various users when their instantaneous channel conditions are relatively favorable. While channel variations are considered to have an adverse impact on constant-rate voice connections, they thus provide the opportunity to improve the throughput of elastic data transfers.

The design and analysis of channel-aware scheduling strategies has attracted tremendous interest over the past several years [4]–[20]. Although the construction of efficient scheduling algorithms for delay-tolerant data transfers is well understood, the extension to delay-sensitive applications adds a dimension of complexity. Exploiting rate variations while satisfying packet-level Quality-of-Service requirements entails major challenges, both in terms of theoretical aspects and practical issues. In particular, the improvement in throughput comes at the expense of a larger variability in the service rate over time, which may be compounded by temporal correlations in the channel quality. While the increase in variability hardly affects delay-tolerant data transfers, it does have a severe impact on delay-sensitive applications, and thus the actual scope for performance gains remains unclear.

The impact of channel-aware scheduling strategies on packet delay has been studied in [21]. The authors specifically consider a max-throughput policy where at each scheduling instant the user with the best channel is selected. The delay is defined as the minimum number of scheduling instants that guarantees all n users successfully receive m packets. As n increases, the expected delay seems to converge to $n \log n$, with the convergence being slower as m increases. For delay-sensitive applications, such a delay may be unacceptable.

Scheduling strategies suited to delay-sensitive applications that maintain some notion of channel-aware scheduling, have

received significant attention in the literature [6], [15], [22]–[24]. We will review some of the proposed strategies in Section III-C. We note here that these studies consider infinite backlogged users [23] and/or a fixed number of users [6], [15], [22]–[24].

In the present paper we consider a wireless system supporting a combination of elastic and streaming traffic. Rather than pursuing any specific scheduling algorithm, we analyze the flow-level performance for various generic resource sharing paradigms. Building on the analysis in [25], we first examine a scenario with rate-adaptive streaming users, and show that the flow-level performance may be evaluated by means of a Processor-Sharing type model, with a state-dependent service rate function that captures the properties of the scheduling algorithm. Simulation experiments demonstrate that the analytical results provide surprisingly accurate estimates for flow-level performance measures such as the transfer delays experienced by the elastic flows, as well as the throughputs obtained by the streaming users. The results indicate that channel-aware scheduling offers substantial performance gains, both to the elastic flows and to the streaming users, and also reveal some intriguing non-monotonicity properties. In particular, if the streaming traffic is served in a channel-oblivious manner, then the transfer delays incurred by the elastic traffic may be non-monotone in the offered load. We further investigate a scenario where the streaming sources have an intrinsic rate profile and stringent delay requirements, and observe that channel-aware scheduling brings only marginal performance gains, and may even have a detrimental effect. In fact, we find that little is sacrificed by simply granting absolute priority to the streaming users over the elastic flows.

The remainder of the paper is organized as follows. In Section II we present a detailed model description. In Section III we analyze the long-term throughput for a static user population in various resource sharing scenarios. We use these results in Section IV to evaluate the flow-level performance in a dynamic setting where users come and go over time as governed by finite random service demands. In Section V we discuss the numerical experiments that we conducted to validate the analytical results and quantify the performance gains from channel-aware scheduling. We make some concluding remarks in Section VI.

II. Model description

We focus on a single base station supporting a combination of streaming and elastic traffic. The base station operates in a time-slotted fashion. In each slot, the base station serves at most one of the users, as will be described in more detail below.

There are K classes of streaming traffic. Class-k streaming connections arrive as a Poisson process of rate ν_k, and have generally distributed holding times with mean τ_k. Define $\sigma_k := \nu_k \tau_k$ as the traffic intensity of the k-th streaming class. The time-average transmission rate of class-k connections is D_k. The instantaneous transmission rates of class-k connections vary over time, and the relative fluctuations around the time-average values are distributed as some random variable Z_k. Note that there is no essential loss in assuming the rate attributes to be identical within classes, since one can easily introduce auxiliary classes to capture heterogeneous characteristics.

There are L classes of elastic traffic. Class-l elastic flows arrive as a Poisson process of rate λ_l, and have generally distributed sizes F_l (in bits). The time-average transmission rate of class-l flows is C_l. The instantaneous transmission rates of elastic flows also vary over time, and the relative variations around the time-average values are distributed as some random variable Y, which we assume to be common to all classes. Denote by $B_l := F_l/C_l$ the service requirement of class-l flows (in seconds), with mean $\beta_l := \mathrm{E}[F_l]/C_l$, and define $\rho_l := \lambda_l \beta_l$ as the offered traffic of the l-th elastic class.

Note that the holding times of streaming users, when admitted into the system, do not depend on the amount of service received, and are thus independent of the level of congestion. In contrast, the elastic flows do not leave the system until the cumulative amount of service received equals their size, so that their delays *do* depend on the degree of competition for service.

III. Static user population

We first consider the (normalized) throughput of the elastic traffic for a static population of $\bar{m} = (m_1, \ldots, m_K)$ streaming users and $\bar{n} = (n_1, \ldots, n_L)$ elastic flows. By normalized throughput, we mean the rate at which the normalized amount of work in the system is reduced per unit time. The normalized amount of work is simply the sum of the remaining service requirements of the active users (in seconds), i.e., the sum of the remaining flow sizes normalized by the time-average transmission rates of the corresponding users (in bits per second). In the next section, we will use these results in order to analyze the flow-level performance in a dynamic context where flows come and go over time as governed by random, finite-duration/size service demands. Evidently, the throughput of the elastic traffic depends on exactly how the transmission resources are shared between the streaming users and elastic flows, and in particular on how time slots are assigned. Below we will determine the throughput in various scenarios of interest.

A. Channel-oblivious scheduling

We first assume that each of the class-k streaming users receives a fraction $\alpha_k(\bar{m}, n)$ of the time slots, with $n := \sum_{l=1}^{L} n_l$, irrespective of the actual channel conditions. The remaining time slots are fairly shared among the elastic flows, also regardless of the feasible transmission rates. Thus the (normalized) throughput of the elastic traffic is

$$H_{\bar{m}}(n) = 1 - \sum_{k=1}^{K} m_k \alpha_k(\bar{m}, n),$$

and the long-term throughput obtained by each of the class-k streaming users is $T_k(\bar{m}, n) = \alpha_k(\bar{m}, n) D_k$. One example

is $\alpha_k(\bar{m},n) = r_k/D_k$, where r_k represents a long-term throughput requirement. A further example is $\alpha_k(\bar{m},n) = w_k/(\sum_{l=1}^{K} w_l m_l + n)$, modeling weighted sharing for adaptive streaming traffic, which reduces to fair sharing for $w_k = 1$ for all $k = 1,\ldots,K$.

Next, we assume that each of the class-k streaming users still receives a fraction $\alpha_k(\bar{m},n)$ of the time slots, irrespective of the actual channel conditions, but that the remaining slots are divided among the elastic flows in a channel-aware manner. Specifically, the elastic flows are allocated time slots according to a weight-based scheduling strategy, where class-l flows are assigned a weight $w_l = 1/C_l$, reciprocal to their time-average transmission rate. A weight-based strategy allocates time slot t to user i^* identified by the largest product of $w_l R_{l,i}(t)$, where $R_{l,i}(t) \equiv C_l Y_{l,i}(t)$ is the feasible rate of the i-th class-l user at time t, and $Y_{li}(t)$ i.i.d. copies of the generic random variable Y with $\mathrm{E}[Y] = 1$. As proved in [6], [9], [25], the class of weight-based scheduling strategies is throughput-optimal, in the sense that any achievable throughput vector can be achieved for suitably chosen weights. As argued in [25], the Proportional Fair scheduling algorithm for the 1xEV-DO system behaves approximately like the above-described weight-based scheduling strategy which assigns weights $w_l = 1/C_l$ to class-l flows. Then,

$$H_{\bar{m}}(n) = (1 - \sum_{k=1}^{K} m_k \alpha_k(\bar{m},n)) G(n),$$

with

$$\begin{aligned} G(n) &= \mathrm{E}\left[\max_{l=1,\ldots,L} \max_{i=1,\ldots,n_l} w_{l,i} R_{l,i}\right] \\ &= \mathrm{E}\left[\max_{l=1,\ldots,L} \max_{i=1,\ldots,n_l} Y_{l,i}\right] = \mathrm{E}\left[\max_{j=1,\ldots,n} Y_j\right], \end{aligned} \tag{1}$$

with $Y_1,\ldots,Y_n$ i.i.d. copies of the generic random variable Y. For example, if Y has an exponential distribution, then $G(n) = \sum_{j=1}^{n} \frac{1}{j}$. The latter assumption is roughly valid when the users have independent Rayleigh fading channels and the feasible rate is approximately linear in the SNR (signal-to-noise ratio). The latter approximation is reasonably accurate when the SNR is not too high. In case $\alpha_k(\bar{m},n) = 1/(m+n)$, with $m := \sum_{k=1}^{K} m_k$, we obtain

$$H_{\bar{m}}(n) = nG(n)/(m+n) \tag{2}$$

and

$$T_k(\bar{m},n) = D_k/(m+n). \tag{3}$$

B. Channel-aware scheduling

We now assume that the streaming users are allocated slots in a channel-aware manner as well. Specifically, class-k streaming users are assigned weights $v_k(\bar{m},n)$. In that case, the throughput $H_{\bar{m}}(n)$ of the elastic traffic may be formally expressed as

$$\mathrm{E}\left[\max_{j=1,\ldots,n} Y_j 1_{\left\{\max_{j=1,\ldots,n} Y_j \geq \max_{k=1,\ldots,K} \max_{i=1,\ldots,m_k} v_{ki} D_k Z_{ki}\right\}}\right],$$

while the long-term throughput $T_k(\bar{m},n)$ of each of the class-k streaming users is

$$\mathrm{E}\left[D_k Z_k 1_{\left\{v_k D_k Z_k \geq \max\{Y_1,\ldots,Y_n,\ \max_{\substack{k=1,\ldots,K \\ i=1,\ldots,m_k}} v_{ki} D_k Z_{ki}\}\right\}}\right].$$

In case $v_k(\bar{m},n) = 1/D_k$ for all $\bar{m},n$, and $Z_k \stackrel{d}{=} Y$ for all $k = 1,\ldots,K$, the above expressions reduce to

$$H_{\bar{m}}(n) = \frac{n}{m+n} G(m+n), \tag{4}$$

and

$$T_k(\bar{m},n) = \frac{D_k}{m+n} G(m+n), \tag{5}$$

respectively.

C. Queue-sensitive scheduling

Queue- or delay-based scheduling policies have been widely proposed for scheduling streaming users [6], [15], [22], [23]. In [22] a feasible Earliest Due Date (FEDD) policy is proposed where at each slot the scheduler chooses the user with the earliest deadline among users in a good channel state. The authors observe that this policy is not always throughput-optimal. The Modified Largest Weighted Delay First (M-LWDF) policy proposed in [6] schedules user i^* such that $i^* = \arg\max_i \gamma_i [W_i(t)]^\beta \mu_i(t)$ where $W_i(t)$ is the head-of-the-line packet delay for user i at time t, $\mu_i(t)$ is its data rate at time t and γ_i and β are arbitrary positive constants. The authors recommend a value of $\gamma_i = a_i/\bar{\mu}_i$ for user i where the a_i is a weight that may be based on delay requirements, and $\bar{\mu}_i$ is the user's long-term average data rate. This policy tries to balance weighted delays, and for the given choice of γ_i, reduces to a channel-aware scheduling strategy when the users are otherwise equal. The exponential rule introduced in [6] and analyzed in [15] schedules the user $i^* = \arg\max_i \gamma_i \mu_i(t) \exp\left(\frac{a_i W_i(t) - \overline{aW}}{1+\sqrt{\overline{aW}}}\right)$ where $\overline{aW} = \frac{1}{N}\sum_i a_i W_i(t)$, and N is the total number of users. This rule tends to equalize weighted delays when the differences are large, and falls to the Proportional Fair policy when the differences are small. In [23] the authors propose a utility-based scheduling algorithm, where the users are scheduled such that the total utility is maximized. The utility for a given user is defined as some concave function of the head-of-line packet delay. Large-deviations results for opportunistic queue-based scheduling policies may be found in [24].

While all these policies tend to optimize some given objective, the amount of information required at the scheduler, such as packet delays, queue lengths, weighted sum of packet delays, etc., may make these policies impractical. Furthermore,

all these papers consider a fixed number of users, whereas in a realistic scenario users come and go as they commence and complete finite service demands. In such settings user-level performance measures such as long-term throughputs and transfer delays become more relevant. Note also that the same scheduling metric, some function of packet delays or queue backlogs, is applied to all users, whereas elastic users are delay-tolerant.

A modification to such scheduling policies where delay-tolerant elastic users are not scheduled based on queue lengths, is to choose at time t, a user $i^*(t)$ as follows:

$$i^*(t) = \arg \max_{i=1,\ldots,N} \begin{cases} w_i R_i(t) & i \text{ is an elastic user,} \\ v_i R_i(t) f(Q_i(t)) & i \text{ is a streaming user,} \end{cases}$$

where $N = m + n$, the weights w_i and v_i are as discussed above, $Q_i(t)$ is the queue length of user i, and for each user $f(\cdot)$ is some function of the queue length. When streaming users have large queues, such a policy gives them priority over elastic users. As their queue lengths become small, the scheduling falls back to a channel-aware scheduling strategy.

The throughput $H_{\bar{m}}(n)$ of the elastic traffic is:

$$\mathrm{E}\left[\max_{j=1,\ldots,n} Y_j 1_{\left\{ \max_{j=1,\ldots,n} Y_j \geq \max_{k=1,\ldots,K} \max_{i=1,\ldots,m_k} Z_{ki} f(Q_{ki}) \right\}} \right].$$

D. Priority scheduling

When the streaming users have an intrinsic rate profile, a simpler policy is one that gives strict priority to streaming users, regardless of the channel conditions. This policy serves streaming users as long as there are packets in their queues, and serves elastic users in a channel-aware manner once streaming users have been served. The normalized throughput of the elastic traffic is then:

$$H_{\bar{m}}(n) = (1 - \sum_{k=1}^{K} m_k \frac{r_k}{D_k}) G(n). \tag{6}$$

The numerical experiments in Section V show that the performance of elastic traffic under this simple policy hardly suffers as compared to that with a queue-based policy.

IV. Flow-level performance

In the previous section we considered the (normalized) throughput of the elastic traffic for a static population of $\bar{m} = (m_1, \ldots, m_K)$ streaming users and $\bar{n} = (n_1, \ldots, n_L)$ elastic flows. We showed that in various scenarios the throughput of the elastic traffic could be described (or approximated) by some function $H_{\bar{m}}(n)$, which depends on $\bar{n}$ through the total number of elastic flows $n = n_1 + \cdots + n_L$ only. The exact nature of the function $H_{\bar{m}}(n)$ and the dependence on $\bar{m}$ and n is determined by the detailed mechanics of the resource sharing at the slot level.

We now use these results in order to analyze the flow-level performance in a dynamic setting where flows come and go over time as governed by random finite-duration/size service demands as described in Section II. We assume that the length of the time slots is short relative to the duration/size and arrival frequency of the service demands. Thus, the resource sharing at the slot level operates on an extremely fast time scale compared to the flow dynamics, making it natural to analyze the flow-level performance in continuous rather than discrete time. The continuous-time, dynamic setting naturally inherits its service characteristics from the discrete-time, static scenario. Specifically, we assume that the instantaneous service rates for any given user population in the dynamic setting coincide with the long-term throughputs for that population in a static scenario.

A. Streaming traffic

We first consider the streaming traffic. Recall that the holding times of streaming users, when admitted into the system, do not depend on the amount of service received, and are thus not affected by the presence of the elastic traffic. We additionally assume that streaming traffic is admitted as long as the resulting configuration of streaming users remains 'admissible'. Specifically, let us suppose that the set of admissible configurations is given by $S \subseteq \mathcal{N}^K$. Note that this set may depend on the number of elastic flows. If we consider no admission control for elastic flows, then in order to avoid instability, $S = \{\bar{m} : \rho < H^*_{\bar{m}}\}$, where $H^*_{\bar{m}} = \sup_{n=1,2,\ldots} H_{\bar{m}}(n) = \lim_{n\to\infty} H_{\bar{m}}(n)$. If we impose admission control on elastic flows as well, then the admissible region for streaming users will depend on the number of elastic flows. Admission control for elastic flows may be based on a maximum number of flows or some minimum rate requirement. In the case of admission control for elastic flows based on a minimum data rate, we have $S = \{\bar{m} : T_k(\bar{m}, n) \geq c^{\mathrm{str}}_{\min}, k = 1, \ldots, K, \frac{H_{\bar{m}}(n)}{n} C_l \geq c^{\mathrm{el}}_{\min}, l = 1, \ldots, L\}$, where $c^{\mathrm{str}}_{\min}$ and $c^{\mathrm{el}}_{\min}$ are minimum data rates for streaming and elastic flows, respectively. We refer to [26] for an extensive discussion of integrated admission control in a wireline context. Then the configuration of streaming users evolves as the population of customers in a loss system with state space S and offered traffic $\sigma_1, \ldots, \sigma_K$ of the various classes. In particular, the number of streaming users of the various classes has a K-dimensional truncated Poisson distribution with parameters $\sigma_1, \ldots, \sigma_K$:

$$\pi_{\bar{m}} = \Pr\left[(M_1, \ldots, M_K) = (m_1, \ldots, m_K)\right] = H_S^{-1} \prod_{k=1}^{K} \mathrm{e}^{-\sigma_k} \frac{\sigma_k^{m_k}}{m_k!},$$

$\bar{m} = (m_1, \ldots, m_K) \in S$, with normalization constant

$$H_S = \sum_{\bar{m} \in S} \prod_{k=1}^{K} \mathrm{e}^{-\sigma_k} \frac{\sigma_k^{m_k}}{m_k!}.$$

B. Elastic traffic

We now turn to the elastic traffic. We first consider the flow-level performance in the presence of a static population $\bar{m} = (m_1, \ldots, m_K)$ of streaming users. As mentioned above, we assume that the instantaneous service rates in the dynamic setting coincide with the long-term throughputs in a static scenario. Thus, each of the elastic flows receives service at rate $H_{\bar{m}}(n)/n$ when there are n elastic flows in total. We additionally assume that elastic traffic is admitted as long as the total number of elastic flows n does not exceed a certain threshold $U_{\bar{m}}$, which may depend on the configuration of streaming users $\bar{m}$. Thus, the configuration of elastic flows in the dynamic setting behaves as the population of customers in a Processor-Sharing system with arrival

rates $\lambda_1, \ldots, \lambda_L$, generic service requirements $B_1, \ldots, B_L$, admission threshold $U_{\bar{m}}$ and state-dependent service rate $G_{\bar{m}}(n)$. Let $(N_{\bar{m},1}, \ldots, N_{\bar{m},L})$ be a random vector representing the number of elastic flows of the various classes at an arbitrary epoch. Denote by $N_{\bar{m}} := N_{\bar{m},1} + \ldots, N_{\bar{m},L}$ the total number of elastic flows in the system. Given that there are n_l elastic class-l flows in the system, let $B^r_{l,i}$ be the remaining normalized service requirement of the i-th class-l flow. The flow-level performance of the elastic traffic then follows from standard results in [27], [28].

Proposition 1: In case $\rho < H^*_{\bar{m}}$ or $U_{\bar{m}} < \infty$,

$$\Pr\left[N_{\bar{m},l} = n_l, B^r_{\bar{m},l,j} \leq t_{l,j}; j = 1, \ldots, n_l, l = 1, \ldots, L\right] = J^{-1}_{\bar{m}} \frac{n! \rho^n}{\phi_{\bar{m}}(n)} \prod_{l=1}^{L} \frac{1}{n_l!} \left(\frac{\rho_l}{\rho}\right)^{n_l} \prod_{j=1}^{n_l} B^r_l(t_{l,j}),$$

with $n = n_1 + \cdots + n_L \leq U_{\bar{m}}$, $\phi_{\bar{m}}(n) := \prod_{i=1}^{n} H_{\bar{m}}(i)$, and normalization constant

$$J_{\bar{m}} := \sum_{n=0}^{U_{\bar{m}}} \frac{\rho^n}{\phi_{\bar{m}}(n)}. \tag{7}$$

In particular,

$$\Pr\left[N_{\bar{m}} = n\right] = J^{-1}_{\bar{m}} \frac{\rho^n}{\phi_{\bar{m}}(n)}, \tag{8}$$

$$\mathrm{E}\left[N_{\bar{m}}\right] = J^{-1}_{\bar{m}} \sum_{n=1}^{U_{\bar{m}}} \frac{n\rho^n}{\phi_{\bar{m}}(n)}, \tag{9}$$

$$\mathrm{E}\left[N_{\bar{m},l}\right] = \frac{\rho_l}{\rho} \mathrm{E}\left[N_{\bar{m}}\right],$$

and the blocking probability is given by

$$p_{\bar{m}} = \Pr\left[N_{\bar{m}} = U_{\bar{m}}\right]. \tag{10}$$

Using Little's law, we find that the mean transfer delay experienced by a class-l elastic flow is given by

$$\mathrm{E}\left[S_{\bar{m},l}\right] = \frac{\beta_l}{\rho(1 - p_{\bar{m}})} \mathrm{E}\left[N_{\bar{m}}\right]. \tag{11}$$

The above formula reflects the celebrated insensitivity property of the Processor-Sharing discipline, which shows that the mean delay of a class-l flow only depends on the service requirement distribution of class l through its mean β_l. In fact, it may be shown that the conditional expected delay of any flow with actual service requirement b is given by

$$\mathrm{E}\left[S_{\bar{m}} | B = b\right] = \frac{b}{\rho(1 - p_{\bar{m}})} \mathrm{E}\left[N_{\bar{m}}\right].$$

Thus, the expected transfer delay incurred by an elastic flow is proportional to its normalized service requirement, with factor of proportionality $\mathrm{E}\left[N_{\bar{m}}\right] / (\rho(1 - p_{\bar{m}}))$. The latter property embodies a certain fairness principle, which means that flows with larger service requirements tend to experience longer delays. Recall that the normalized service requirement encapsulates both the file size and the time-average transmission of a flow, and is expressed in time units rather than data bits.

C. Quasi-stationary regime

In the previous subsection we analyzed the flow-level performance of the elastic traffic in the presence of a static population of streaming users. We now consider a scenario where the configuration of streaming users is no longer static, but also varies over time. As described above, the number of streaming users of the various classes then follows a K-dimensional truncated Poisson distribution with parameters $\rho_1, \ldots, \rho_K$.

In order to obtain a tractable result, we will assume that the flow-level dynamics of the streaming traffic occur on a relatively slow time scale compared to those of the elastic traffic. In the limiting scenario, referred to as *quasi-stationary* regime, a complete separation of time scales occurs, and the elastic traffic will approximately reach some sort of steady state in between changes in the population of streaming users. In that case, the flow-level performance of the elastic traffic may be obtained by weighing that derived in Proposition 1 with the corresponding distribution for the streaming traffic specified above. The quasi-stationary regime will tend to provide an accurate approximation when the typical duration of the elastic flows is significantly shorter than the holding time of the streaming users. We refer to [29] for a similar analysis of an integrated system in a wireline setting.

Proposition 2: In case $\rho < H^*_{\bar{m}}$ or $U_{\bar{m}} < \infty$ for all $m \in S$,

$$\Pr\left[N_l = n_l, B^r_{l,j} \leq t_{l,j}; j = 1, \ldots, n_l, l = 1, \ldots, L\right] = \sum_{\bar{m} \in S} \pi_{\bar{m}} \Pr\left[N_{\bar{m},l} = n_l, B^r_{\bar{m},l,j} \leq t_{l,j}; j = 1, \ldots, n_l, l = 1, \ldots, L\right],$$

with $\pi_{\bar{m}}$ as specified above. In particular,

$$\Pr\left[N = n\right] = \sum_{\bar{m} \in S} \pi_{\bar{m}} J^{-1}_{\bar{m}} \frac{\rho^n}{\phi_{\bar{m}}(n)}, \tag{12}$$

$$\mathrm{E}\left[N\right] = \sum_{\bar{m} \in S} \pi_{\bar{m}} J^{-1}_{\bar{m}} \sum_{n=1}^{U_{\bar{m}}} \frac{n\rho^n}{\phi_{\bar{m}}(n)}, \tag{13}$$

$$\mathrm{E}\left[N_l\right] = \sum_{\bar{m} \in S} \pi_{\bar{m}} \frac{\rho_l}{\rho} \mathrm{E}\left[N_{\bar{m},l}\right],$$

and the blocking probability is given by

$$p = \sum_{\bar{m} \in S} \pi_{\bar{m}} p_{\bar{m}}. \tag{14}$$

D. Fluid regime

As mentioned above, the configuration of streaming users is not influenced by the presence of the elastic traffic, as long as the admission of streaming traffic does not take the elastic flows into consideration. However, the amount of service received by the streaming users *does* depend on the interaction with the elastic traffic.

In order to evaluate the average obtained throughput, we will assume a separation of time scales as above. From the perspective of streaming flows then, the system is in a *fluid*

regime. In this case, the average throughput received by a class-k streaming user may be determined as

$$\mathrm{E}\left[T_k\right] = \sum_{\bar{m} \in S} \pi_{\bar{m}} T_{\bar{m},k}, \tag{15}$$

with $\pi_{\bar{m}}$ as specified above, and

$$T_{\bar{m},k} = \sum_{n=0}^{U_{\bar{m}}} \Pr N_{\bar{m}} = n T_k(\bar{m}, n), \tag{16}$$

and $T_k(\bar{m}, n)$ the long-term throughput received by a class-k streaming user in a static scenario with $\bar{m} = (m_1, \ldots, m_K)$ streaming users and a total number of n elastic flows as derived in Section IV-A. Note that the admissible region for streaming flows, S, may depend on the number of elastic flows as discussed in Section IV-A.

V. Numerical experiments

We now present the numerical experiments that we conducted to validate the analytical results and compare the gains from channel-aware scheduling in terms of flow-level performance in various scenarios as described in the previous sections. We evaluate the mean transfer delay incurred by elastic flows and the mean throughput received by streaming users, for the case of both a fixed number and dynamic population of streaming users. Throughout the mean size of the elastic flows is assumed to be 60 Kbytes. Rate-adaptive streaming users are considered in Sections V-A and V-B, while streaming sources with an On-Off rate profile are studied in Section V-C. In case of a dynamic population of streaming users, the mean holding time is $\tau = 167$ s. Throughout, *channel-oblivious* refers to channel-oblivious scheduling of streaming users and similarly *channel-aware* pertains to channel-aware scheduling of streaming users. Elastic flows are always scheduled in a channel-aware manner.

The system operates in a time-slotted fashion, with a slot duration of 1.67 ms as in the CDMA 1xEV-DO system. The simulations were run for 10,000,000 slots, or about 16,700 seconds of real time. We assume that users have independent Rayleigh fading channels, and consider two scenarios for the distribution of the time-average Signal-to-Noise Ratio (SNR): (i) statistically homogeneous users each with a time-average SNR of 0 dB; (ii) heterogeneous users with a time-average SNR uniformly distributed on $[-5, 7]$ dB. The instantaneous transmission rate of a user as a function of the instantaneous SNR is determined according to the CDMA 1xEV-DO rate table [1].

In case of a fixed number of streaming users m, the analytical estimate for the mean transfer delay of elastic flows is calculated from Equations (7)–(11). The mean throughput of streaming users is computed from Equation (16), with $\Pr\left[N_m = n\right]$ determined by Equation (8). In case of a dynamic population of streaming users, we use the quasi-stationary approximation in Equations (12)–(14) for the mean transfer delay of the elastic flows, and the fluid approximation in Equation (15) for the mean throughput of streaming users. The functions $H_m(n)$ used in determining the coefficients $\phi_m(n)$ and $T(m, n)$ depend on the specific scenario of interest; in the 'channel-oblivious' case, Equations (2) and (3) are invoked, whereas in the 'channel-aware' case, Equations (4) and (5) are used.

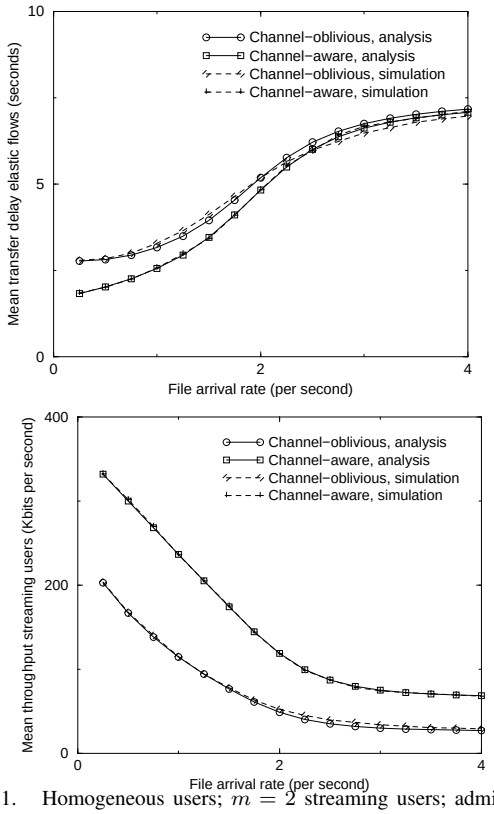

Fig. 1. Homogeneous users; $m = 2$ streaming users; admission threshold $U_2 = 18$.

A. Homogeneous channel statistics

We first consider statistically homogeneous users with a time-average SNR of 0 dB.

Figure 1 displays the mean transfer delay incurred by elastic flows (top) and the mean throughput received by streaming users (bottom) as a function of the arrival rate of elastic flows, for a static population of $m = 2$ streaming users. By scheduling streaming users in a channel-aware manner, the transfer delay experienced by elastic flows increases by a limited amount at low load, and even less so at higher load. The throughput enjoyed by the rate-adaptive streaming users, however, increases quite significantly.

Figure 2 shows results for a static population of $m = 8$ streaming users. We observe that the mean transfer delay is non-monotone in the offered load of the elastic traffic when the scheduling is channel-oblivious among streaming users. This remarkable observation may be explained as follows. As the elastic load and hence the number of active flows go up, two opposite effects occur: (i) the scheduling gain increases (favorable); and (ii) the fraction of time slots per active flow decreases (unfavorable). At low load, the favorable impact dominates the unfavorable effect, since some additional

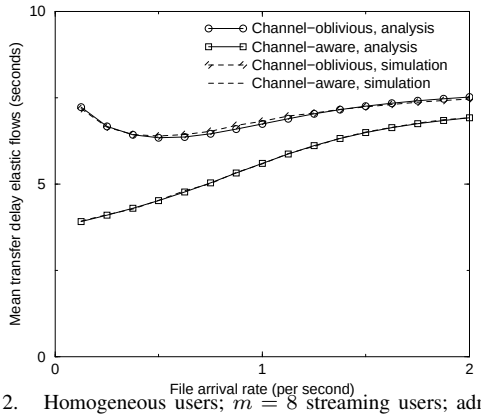

Fig. 2. Homogeneous users; $m = 8$ streaming users; admission threshold $U_8 = 12$.

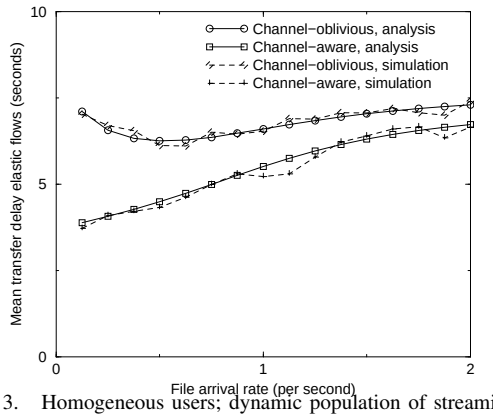

Fig. 3. Homogeneous users; dynamic population of streaming users with a load of $\sigma = 8$; admission threshold $U_2 = 12$, $N_{stream} = 15$.

time slots are taken away from the streaming users, and the function $G(\cdot)$ in (1) rises sharply, reducing the transfer delay. At higher load, the balance reverses, since the elastic flows essentially start competing among themselves, and the function $G(\cdot)$ flattens out, yielding an increase in the transfer delay. Such a phenomenon does not occur when streaming users are also scheduled in a channel-aware manner because the growth in the function $G(\cdot)$ then already levels off at low elastic load.

We now turn to a dynamic population of streaming users. Figure 3 shows that for an offered streaming load of $\sigma = 8$, the throughput obtained by the streaming users increases with channel-aware scheduling. We further observe again the non-monotonicity in the transfer delays experienced by the elastic flows. Also, note that the analytical results provide highly accurate estimates of both the mean transfer delays of the elastic flows and the mean throughputs of the streaming users.

B. Heterogeneous channel statistics

We now consider heterogeneous users with a time-average SNR uniformly distributed on $[-5, 7]$ dB. Figures 4 and 5 show that channel-aware scheduling continues to yield substantial throughput gains for the streaming users, without significantly affecting the elastic flows. The non-monotonic

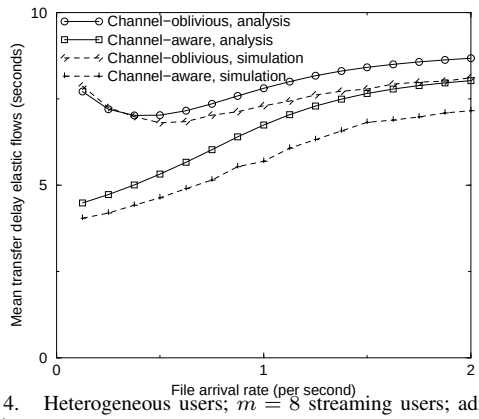

Fig. 4. Heterogeneous users; $m = 8$ streaming users; admission threshold $N(8) = 12$.

behavior of the transfer delays experienced by elastic flows manifests itself again, both for a fixed and a dynamic population of streaming users as in Figures 4 and 5, respectively. The analytical results continue to provide reasonably accurate estimates, despite the fact that the underlying assumption of symmetric relative rate fluctuations no longer strictly applies.

C. Streaming users with an On-Off rate profile

We now consider a scenario where the streaming users are no longer rate-adaptive, but have an intrinsic rate profile. Specifically, the streaming users have queues driven by exponential On-Off sources, each with a peak rate of 32 Kbits/s, mean On-periods of 100 ms, mean Off-periods of 100 ms and a transmit buffer of 800 bytes, corresponding to a delay tolerance of 200 ms. The Rayleigh fading is simulated using Jakes model, with a Doppler frequency of either 5 or 50 Hz.

We first consider the same scheduling strategies as before, with the only change that streaming users are not scheduled when their queues are empty. (In fact, the streaming users are not scheduled when their queues are below a threshold of 480 bits. We note that without such a threshold, the performance may severely degrade due to partial slot filling effects, especially when the streaming users receive priority over the elastic flows.) The analytical estimate for the mean throughput of the elastic flows is obtained using Equation (6). The actual throughput is likely to be somewhat lower, since the streaming users tend to get scheduled at lower-than-average rates and thus take away a higher fraction of the slots when they generate bursty traffic and the channel processes are correlated over time, as is reflected in Figures 8 and 9.

Figure 6 presents the transfer delays incurred by the elastic flows (top) and loss rates suffered by streaming users (bottom) for a Doppler frequency of 50 Hz. Figure 7 graphs similar results for a Doppler frequency of 5 Hz. The performance experienced by the elastic flows is not impacted a great deal by whether the scheduling for streaming users is channel-aware or not, at either Doppler frequencies. At a low Doppler frequency, the loss rate sustained by the streaming users increases significantly with channel-aware scheduling at high

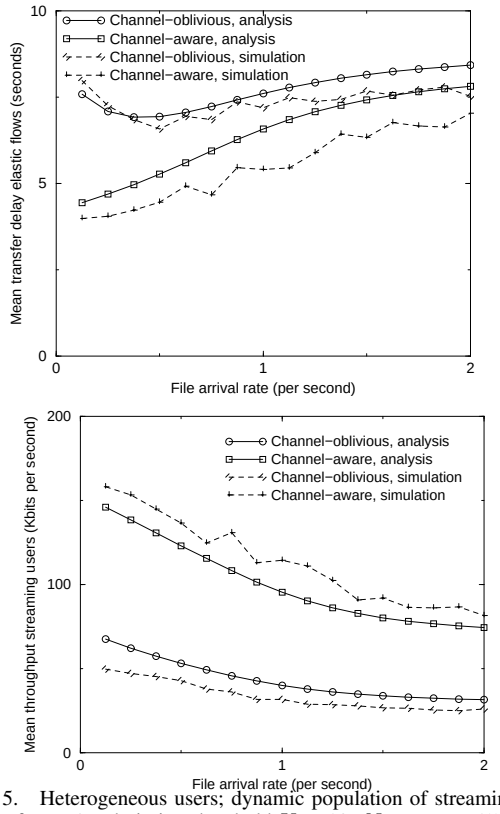

Fig. 5. Heterogeneous users; dynamic population of streaming users with a load of $\sigma = 8$; admission threshold $U = 12$, $N_{stream} = 15$.

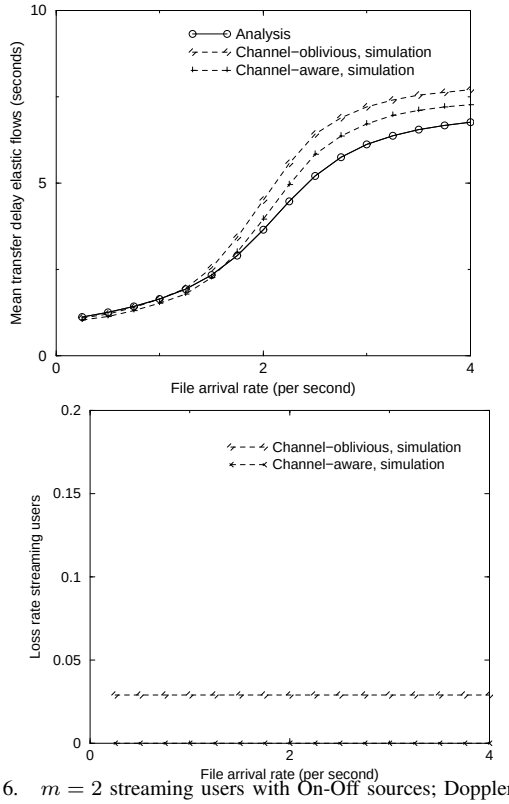

Fig. 6. $m = 2$ streaming users with On-Off sources; Doppler frequency of 50 Hz.

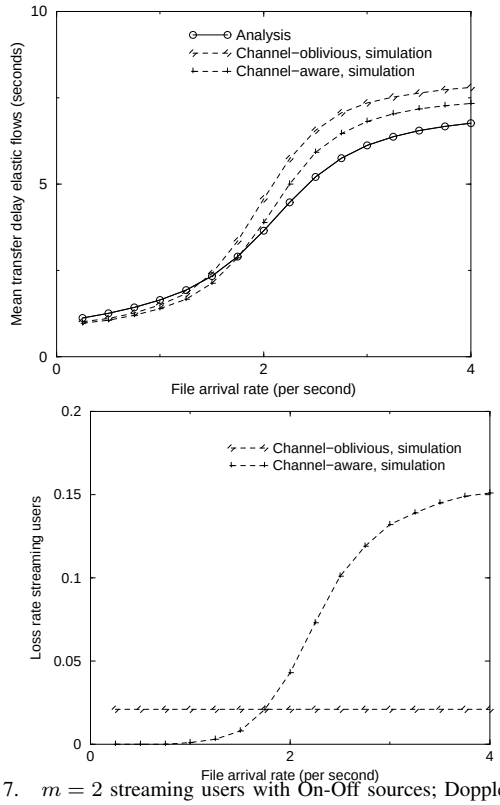

Fig. 7. $m = 2$ streaming users with On-Off sources; Doppler frequency of 5 Hz.

load, due to the longer delays in the presence of finite buffers.

We finally consider queue-based and priority scheduling for streaming users. Figures 8 and 9 depict the mean transfer delays incurrred by elastic flows at Doppler frequencies of 50 Hz and 5 Hz, respectively. For these figures "channel-oblivious + priority" refers to priority scheduling for streaming users without any consideration of channel conditions and "channel-aware + priority" pertains to a weighted-queue-based scheduling for streaming users, where the weight is set to the instantaneous transmission rate. Referring to Section III, the latter means $v_k(\bar{m}, n) = 1/D_k$. We do not show the corresponding results for streaming users here, because they enjoy excellent performance with priority scheduling as evidenced by negligible loss rates throughout.

These results show that the mean transfer delay incurred by elastic flows is not significantly impacted by priority scheduling of streaming users, channel-aware or not. This suggests granting priority to streaming users as a simple operational rule, with channel-aware scheduling of streaming users being of somewhat lesser importance.

VI. Conclusion

We have examined resource sharing paradigms in shared downlink networks such as CDMA 1xEV-DO, with a mix of streaming and elastic traffic. We have analyzed the flow-level performance for various scheduling strategies. Analytic evaluation was compared with extensive simulation results. We

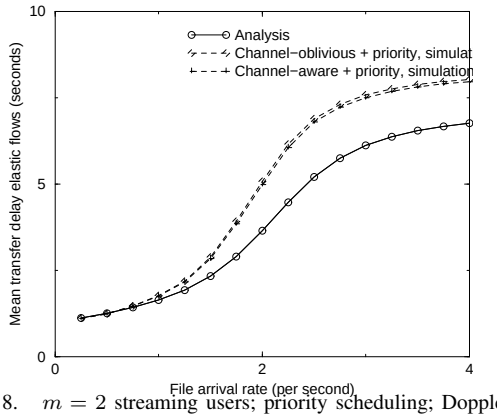

Fig. 8. $m = 2$ streaming users; priority scheduling; Doppler frequency of 50 Hz.

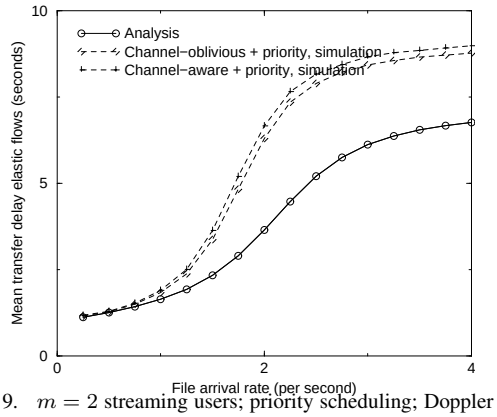

Fig. 9. $m = 2$ streaming users; priority scheduling; Doppler frequency 5 Hz

have observed that channel-aware scheduling offers significant performance gains for elastic users and for rate-adaptive streaming users. For a scenario of streaming users with a given rate profile and correlations in the channel, we demonstrated a detrimental effect on the performance of streaming users, with marginal gains for elastic users. Furthermore, we observed that under a simple scheduling strategy with strict priority to streaming users the elastic users suffered little loss in performance.

REFERENCES

[1] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Commun. Mag.*, vol. 38, no. 7, pp. 70–77, 2000.

[2] E. Chaponniere, P. Black, J. Holtzman, and D. Tse, "Transmitter directed code division multiple access system using path diversity to equitably maximize throughput," US Patent 6,449,490, 2002.

[3] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE VTC 2000 Spring Conf.*, 2000, pp. 1854–1858.

[4] R. Agrawal, A. Bedekar, R. La, and V. Subramanian, "Class and channel condition based weighted proportional fair scheduler," in *Teletraffic Engineering in the Internet Era, Proceedings of ITC-17, Salvador da Bahia*, J. de Souza, N. da Fonseca, and E. de Souza e Silva, Eds. North-Holland, Amsterdam, 2001, pp. 553–565.

[5] R. Agrawal and V. Subramanian, "Optimality of certain channel-aware scheduling policies," in *"Proc. 40th Annual Allerton Conf. Commun., Control, Comp."*, 2002, pp. 1532–1541.

[6] D. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Prob. Eng. Inf. Sc.*, vol. 18, pp. 191–217, 2004.

[7] D. Andrews, L. Qian, and A. Stolyar, "Optimal utility-based multi-user throughput allocation subject to throughput constraints," in *Proc. IEEE INFOCOM*, 2005.

[8] T. Bonald, "A score-based opportunistic scheduler for fading radio channels," in *Proc. European Wireless Conf.*, 2004.

[9] S. Borst and P. Whiting, "Dynamic channel-sensitive scheduling algorithms for wireless data throughput optimization," *IEEE Trans. Veh. Techn.*, vol. 52, pp. 569–586, 2003.

[10] X. Liu, E. Chong, and N. Shroff, "A framework for opportunistic scheduling in wireless networks," *Comp. Netw.*, vol. 41, pp. 451–474, 2003.

[11] H. Kushner and P. Whiting, "Asymptotic properties of proportional fair sharing algorithms," in *Proc. 40th Annual Allerton Conf. Commun., Control., Comp.*, 2002, pp. 1051–1059.

[12] D. Park, J. Seo, H. Kwon, and B. Lee, "Wireless packet scheduling based on the cumulative distribution function of user transmission rates," *IEEE Trans. Commun.*, vol. 53, pp. 1919–1929, 2005.

[13] S. Patil and G. De Veciana, "Managing resources and quality-of-service in heterogeneous wireless systems exploiting opportunism," Submitted for publication, 2005.

[14] ——, "Measurement-based opportunistic scheduling for heterogeneous wireless systems," Submitted for publication, 2005.

[15] S. Shakkottai and A. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real time data in HDR," in *Teletraffic Engineering in the Internet Era, Proceedings of ITC-17, Salvador da Bahia*, J. de Souza, N. da Fonseca, and E. de Souza e Silva, Eds. North-Holland, Amsterdam, 2001, pp. 793–804.

[16] ——, "Scheduling for multiple flows sharing a time-varying channel: the exponential rule," in *American Math. Soc. Transl., Series 2, A volume in memory of F. Karpelevich*, Y. Suhov, Ed., 2002, vol. 207, pp. 185–202.

[17] A. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multi-user throughput allocation," *Operations Research*, vol. 53, pp. 12–25, 2005.

[18] V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel state information for throughput maximization," in *Proc. IEEE INFOCOM*, 2003.

[19] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1277–1294, 2002.

[20] D. Wu and R. Negi, "Downlink scheduling in a cellular network for quality-of-service assurance," *IEEE Trans. Veh. Techn.*, vol. 53, pp. 1547–1557, 2004.

[21] M. Sharif and B. Hassibi, "A delay analysis for opportunistic transmission in fading broadcast channels," in *Proc. IEEE INFOCOM*, 2005.

[22] S. Shakkottai and R. Srikant, "Scheduling real-time traffic with deadlines over a wireless channel," *Wireless Networks*, vol. 8, no. 1, pp. 13–26, 2002.

[23] P. Liu, R. Berry, and M. Honig, "Delay-sensitive packet scheduling in wireless networks," in *Proc. IEEE WCNC 2003*, New Orleans, LA, March 16-20 2003.

[24] L. Ying, R. Srikant, A. Eryilmaz, and G. Dullerud, "A large deviations analysis of scheduling in wireless networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 5088–5098, Nov. 2006.

[25] S. Borst, "User-level performance of channel-aware scheduling algorithms in wireless data networks," in *Proc. IEEE INFOCOM*, 2003.

[26] N. Benameur, S. B. Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J. Roberts, "Integrated admission control for steaming and elastic traffic," in *Proc. QoFIS*, 2001.

[27] J. Cohen, "The multiple phase service network with generalized processor sharing," *Acta Informatica*, vol. 12, pp. 245–284, 1979.

[28] F. Kelly, *Reversibility and Stochastic Networks*. Wiley, 1979.

[29] F. Delcoigne, A. Proutière, and G. Régnié, "Modelling integration of streaming and data traffic," *Perf. Eval.*, vol. 55, no. 3–4, pp. 185–209, 2004.