Throughput and Delay Performance of DSL Broadband Access with Cross-layer Dynamic Spectrum Management

Paschalis Tsiaflakis, Yung Yi, Mung Chiang, and Marc Moonen

Abstract

DSL broadband access suffers from crosstalk among different lines within the same cable bundle. Dynamic spectrum management (DSM) refers to a set of techniques to mitigate the impact of crosstalk leading to spectacular performance gains. DSM research has mainly aimed at physical layer performance metrics, such as data rates and transmit powers. However, for many applications higher-layer performance metrics, such as throughput and delay, may be much more important to improve user satisfaction. In this paper, we provide a cross-layer DSM framework to study throughput and delay performance by looking at scheduling and DSM together. We show how optimal scheduling can be combined with both optimal and suboptimal DSM and provide throughput-optimal scheduling algorithms which require only polynomial complexity. We analytically study the impact on delay performance of achieving throughput-optimality with suboptimal DSM compared to optimal DSM. We then present extensions that significantly improve delay performance by exploiting the specific structure of the problem, such as the temporal-spectral correlation property. Furthermore, we propose a second cross-layer DSM framework that achieves throughput-optimal scheduling with suboptimal DSM, but in addition also significantly reduces overall power consumption. Finally, we analyze and quantify the tradeoff between throughput, delay and power consumption for concrete DSL scenarios.

Keywords: Digital Subscriber Line, Dynamic Spectrum Management, Scheduling, Throughput-Optimality, Energy-Efficiency

I. Introduction

Digital subscriber line (DSL) technology refers to a family of technologies that provide digital broadband access over the local telephone network. It is currently the most popular wireline broadband access technology with a global market share of 63%, corresponding to more than 330 million DSL subscribers [2]. The main

P. Tsiaflakis and M. Moonen are with the EE. Dept. (ESAT-SCD), KU Leuven, Belgium (e-mail: {paschalis.tsiaflakis, marc.moonen}@esat.kuleuven.be). Y. Yi is with the Dept. of Electr. Eng., KAIST (Korea Advanced Institute Science and Technology), South Korea (e-mail: yiyung@kaist.edu). M. Chiang is with the Dept. of Electr. Eng., Princeton University, USA (e-mail:chiangm@princeton.edu). Paschalis Tsiaflakis is a postdoctoral fellow funded by the Research Foundation - Flanders (FWO). This research work was carried out in the frame of K.U.Leuven Research Council CoE EF/05/006 OPTEC and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, and the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, 2007-2011). This work was in part supported by AFOSR MURI grant FA9550-09-1-0643 and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0015042). Part of this work was presented at IEEE Globecom, 2008 [1].

reason for its popularity is its low deployment cost, as DSL reuses the twisted pairs of the existing telephone network infrastructure to connect the subscribers to the Internet backbone.

One of the major impairments that limits further improvement of DSL performance, is crosstalk, i.e., the electromagnetic interference amongst different lines (i.e., users) in the same cable bundle. The presence of crosstalk transforms the DSL network into a very challenging multi-user multi-carrier interference network, in which the transmission of one user can significantly impact the transmission of all other users. One promising set of techniques to tackle this crosstalk problem and to significantly boost performance, is referred to as dynamic spectrum management (DSM). DSM consists of two main approaches: spectrum coordination and signal coordination. In spectrum coordination, the users' transmit spectra are jointly optimized so as to prevent the impact of crosstalk [3]. Signal coordination, also referred to as vectoring, consists of jointly processing the transmitted or received signals so as to actively cancel the impact of crosstalk [4]. In this paper, we focus on spectrum coordination, also referred to as spectrum management, spectrum balancing, or multi-carrier power control. From an information-theoretic point of view, the spectrum coordination scenario can be considered as a multi-carrier interference channel where each user treats the interference from the other users as noise. Many DSM algorithms[1] have been proposed in literature to address the spectrum coordination problem, ranging from fully autonomous[2] [5], [6] and distributed[3] [7], [8] to centralized algorithms[4] [9]–[11].

Research on DSM algorithms has mainly aimed at physical layer performance metrics, e.g. at maximizing the aggregate data rates subject to power constraints, or, recently also, at minimizing the aggregate transmit powers subject to minimum data rate constraints [12]–[14]. However, considering actual applications over DSL networks such as real-time data (e.g. video and voice) or elastic data (e.g. Internet service) delivery, it is crucial to understand the more direct impact of DSM algorithms on the user-perceived throughput or delay performance over a longer time-scale. This requires an extension of the standard physical layer DSM framework, as typically used in DSM literature [5]–[11], to a cross-layer DSM framework that allows to study and analyze time-dynamic behaviour and performance metrics, such as throughput and delay. In this framework, users generate bursty data traffic and so have a finite instantaneous workload rather than an infinite workload (as assumed by typical DSM algorithm design). The DSL network under bursty data traffic is then modeled as a constrained queueing system, constrained due to the crosstalk. Note that the physical

[1]In the rest of the text DSM refers to spectrum coordination.

[2]Fully autonomous DSM algorithms are DSM algorithms in which each user chooses its transmit powers autonomously based on locally available information only.

[3]Distributed DSM algorithms are DSM algorithms in which each user chooses its transmit powers based on locally available information as well as information obtained from other users through limited message-passing.

[4]Centralized DSM algorithms are DSM algorithms where the transmit powers of all users are determined in a centralized location such as the spectrum management center (SMC), where one has access to full knowledge of the channel environment.

layer DSM framework provides infinitely many non-comparable points on the boundary of the achievable rate region, as obtained with current DSM algorithms. However, it is not clear which point should then be picked in the operation of a given DSL network under a given traffic load. As will become clear, the cross-layer DSM framework will provide a strategy for picking the most suitable operational point at each time instant.

We would like to remark here that DSL standards currently only allow a limited time-dynamic control by mechanisms such as bit swapping and seamless rate adaptation. The proposed cross-layer DSM framework, however, demonstrates how the time dimension could be exploited more efficiently to optimize throughput and delay performance, and so provides a motivation for developing more powerful control mechanisms.

In this paper, we investigate how data rate scheduling and physical layer DSM can be combined in the proposed cross-layer DSM framework. Both optimal scheduling and DSM are computationally intractable. We address the question of how relaxations on both parts work well together towards a practical joint rate scheduling and DSM design with good throughput and delay performance. For this we exploit the specific structure of DSL DSM problems and algorithms, which involves (i) a typical discrete definition of global optimality, (ii) the availability of powerful DSM algorithms whose performance depends on the chosen initial points and that are monotonically increasing, and (iii) the temporal-spectral correlation property. We also investigate how this cross-layer DSM framework can be extended to reduce power consumption, and discuss the impact of such ‘greening’ on throughput and delay. Related work that highlights the possibility of applying DSM through spectrum or signal coordination in a cross-layer setting is reported in [15]–[17]. In [15] it is mentioned that QPS-scheduling [16] which is elaborated for the fading wireless broadcast channel, can be applied to the DSM setting. In [17] a joint scheduling and partial crosstalk cancellation solution is presented for DSL, thus focusing on signal coordination rather than spectrum coordination.

An outline and the main contributions of the paper are as follows:

In Section II, we provide the cross-layer DSL system model, with related performance metrics. In Section III, we provide a first cross-layer DSM framework, motivated by recent research advances in other areas, e.g., wireless networks and switching systems. This framework provides tools to understand the throughput and delay performance of DSM algorithms applied at the physical layer, as well as gives us practical implications on the design of future DSM algorithms. The proposed framework facilitates the characterization of throughput and delay properties, and also discloses the generic trade-off between complexity, and throughput or delay. Using this framework, we then connect throughput-optimal scheduling with globally optimal DSM algorithms, which have an intractable computational complexity.

In Section IV, we show that, somewhat surprisingly, it is possible to achieve an optimal throughput performance by using (randomized) sub-optimal DSM algorithms which require only polynomial time complexity.

However, the price to pay is quantified as increased delay. We then provide algorithms that significantly improve the delay performance with only small extra complexity, by exploiting the specific structure of our problem, namely *the temporal-spectral correlation (TSC) property,* and also FDMA optimality for large crosstalk DSL scenarios.

In Section V, we extend the throughput-optimal algorithms to a power-efficient setting, which sustains throughput-optimality with less power consumption. The starting point is to exploit the power-efficient DSM algorithms of [12], [13] that explicitly consider the power consumption in the objective function in conjunction with the aggregate data rate. We present this power-efficient cross-layer DSM framework as a slight variant of the cross-layer DSM framework of Section III, for which similar ideas to improve delay performance can then be directly applied. However, we show that throughput-optimality with consideration of power-efficiency, comes at the cost of increased delay.

Finally, in Section VI we evaluate the proposed throughput-optimal scheduling algorithms for DSL scenarios with realistic system and channel settings. This allows us to quantify the trade-offs between throughput, delay, average data rates and average power consumption, and to demonstrate the potential of the proposed cross-layer DSM setting.

II. System Model and Performance Metrics

A. System Model

Network and Traffic Models. We consider a discrete time slotted system, indexed by t, consisting of N interfering DMT (Discrete Multi-Tone)-DSL modems or users. We denote by K the number of frequency bands or tones available for each user. We abuse the notations N and K to refer to the index set of users and tones. Each user has an infinite-size buffer which is fed by exogenous arrivals. We denote by $A^n(t)$ the number of arrivals (in bits/slot) to user $n \in N,$ at time $t.$ Note that we use the suffix superscript 'n' to denote the user index throughout the whole text. The arrival process is assumed to be i.i.d. across time slots, where $\mathbb{E}[A^n(t)] = \lambda^n$.[5] We assume that the duration of one time slot is small enough so that λ^n is upper bounded by some constant, i.e., $A^n(t) \leq A_{\max},$ a.s. (almost surely), $\forall n \in N,\ \forall t \geq 0.$ We denote by $\boldsymbol{\lambda} = (\lambda^n :\ \ n \in N)$ the (mean) arrival rate vector to the system. We remark that the ideas and analyses in this text hold for any choice of time slot duration, i.e. also for those scenarios where a longer time slot duration is needed due to, e.g., message passing overheads.

Resource Model. The network resources are represented by a finite set $\mathcal{R}$ of feasible rate vectors, referred to as the *achievable rate region* describing the simultaneously achievable rates (in bits/slot) of the users. The data rates of the users in turn depend on their transmit powers and the resulting interference across tones and users. We first introduce notation, and then characterize the achievable rate region $\mathcal{R}.$

[5]The results of this paper can be readily extended to the case with correlated arrivals under mild technical conditions.

- s_k^n and $s_k^{n,\text{mask}}$ are the transmit power and the spectral mask constraint for user n on tone k, and $\boldsymbol{s} = (s_k^n : n \in N, k \in K)$ is the vector that contains the transmit powers of all users over all tones,
- $b_k^n(\boldsymbol{s})$ is the bit rate of user n on tone k,
- P^n is the total power budget available to user n,
- $[\mathbf{H}_k]_{n,m} = h_k^{n,m}$ represents an $N \times N$ matrix containing the squared magnitude of the channel gains from transmitter m to receiver n on tone k, where the diagonal elements are the direct channels and the off-diagonal elements are the crosstalk channels,
- σ_k^n is the noise power for user n on tone k, which contains thermal noise, alien crosstalk and radio frequency interference (RFI),
- Γ is the signal-to-noise ratio (SNR) gap to capacity, which is a function of the desired bit error ratio (BER), the coding gain and noise margin [18],
- f_s is the DMT symbol rate.

Using the above notations, $b_k^n(\boldsymbol{s})$ is given by:

$$b_k^n(\boldsymbol{s}) \triangleq \log_2\left(1 + \frac{1}{\Gamma}\frac{h_k^{n,n}s_k^n}{\sum\limits_{m\in N, m\neq n} h_k^{n,m}s_k^m + \sigma_k^n}\right) \text{ bits/ s/ Hz.} \tag{1}$$

Then, the achievable rate region is characterized as:

$$\mathcal{R} = \left\{(R^n : n \in N) | R^n = f_s \sum_{k\in K} b_k^n,\ \forall n \in N : \sum_{k\in K} s_k^n \leq P^n, \forall n \in N, k \in K :\ 0 \leq s_k^n \leq s_k^{n,\text{mask}}\right\}. \tag{2}$$

We denote by R_{max} an upper bound on the achievable rate for any user over all possible transmit power allocations. We also define $\Omega_1 = A_{\text{max}} + R_{\text{max}}$ and $\Omega_2 = A_{\text{max}}^2 + R_{\text{max}}^2$, which are constants that will be used later. Note that $\mathcal{R}$ can be considered to be a convex set when the number of tones is large [9], [19], [20].

The transmit powers that are actually used in practical systems are discretized. Denote by $\hat{\mathcal{R}}$ the rate region achieved with the transmit powers discretized up to Δ accuracy, i.e.,

$$\hat{\mathcal{R}} = \left\{(R^n : n \in N) | R^n = f_s \sum_{k\in K} b_k^n,\ \forall n \in N : \sum_{k\in K} s_k^n \leq P^n,\ \forall n \in N, k \in K : s_k^n \in \mathcal{D}_k^n\right\}, \tag{3}$$

$$\text{with} \quad \mathcal{D}_k^n \triangleq \{0, \Delta, 2\Delta, \ldots, \Delta^{\max}\}, \tag{4}$$

where $\Delta^{\max} = \Delta \times \text{floor}(\min(P^n/\Delta, s_k^{n,\text{mask}}/\Delta))$ denotes the maximum integer multiple of Δ smaller than the minimum of the corresponding spectral mask and total power budget. We also define set $\mathcal{D}_k$, which will be used later, as follows,

$$\mathcal{D}_k = \{(y_k^n : n \in N) | \forall n \in N : y_k^n \in \mathcal{D}_k^n\}. \tag{5}$$

Scheduling Algorithm. A scheduling algorithm chooses a *rate schedule* $(\boldsymbol{R}(t) = (R^n(t) : n \in N))_{t=0}^{\infty}$, $\boldsymbol{R}(t) \in \mathcal{R}$ over time. Note that the rate schedule $(\boldsymbol{R}(t))_{t=0}^{\infty}$ is determined by the allocated transmit power

spectra $(s_k^n(t) : n \in N, k \in K)_{t=0}^{\infty}$.

Queueing Dynamics. The evolution of the per-user queue lengths is governed by random arrivals as well as the adopted scheduling algorithm. Denote by $Q^n(t)$ the queue length of user n at time t. The queueing dynamics are represented by the following recursion: for all users $n \in N$,

$$Q^n(t+1) \;=\; \left[Q^n(t) - R^n(t)\right]^+ + A^n(t+1), \tag{6}$$

where $[x]^+ = \max(x, 0)$, and $R^n(t)$ is determined by the scheduling algorithm. We also define $\boldsymbol{Q}(t) = (Q^n(t), n \in N)$.

B. Performance Metrics

(a) Throughput. We first define the notion of stability, which essentially represents the condition that queue lengths remain finite.

Definition 2.1 (Stability): A system is said to be stable for a given rate schedule, if the aggregate queue lengths are kept bounded, i.e., $\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\left[\sum_{n\in N} Q^n(t)\right] < \infty$.

A performance objective of any scheduling algorithm is to guarantee stability whenever possible, i.e., whenever the given arrival rate vector $\boldsymbol{\lambda}$ belongs to the throughput-region defined as follows:

Definition 2.2 (Throughput-region): The throughput-region $\Lambda \subset \mathbb{R}_+^N$ is the set of all arrival rate vectors $\boldsymbol{\lambda}$ for which there exists a rate schedule stabilizing the system.

No rate schedule can stabilize the system for an arrival rate vector outside the throughput-region Λ. The throughput-region can be characterized as: Λ is the *largest open set* included in the convex-hull$(\mathcal{R})$ based on the classical time-sharing argument. We say that a scheduling algorithm that can stabilize the system for any arrival rate vector in Λ is *throughput-optimal*. We define the boundary of the throughput-region as:

Definition 2.3 (Boundary of throughput-region): The boundary of the throughput-region Λ^b is the set of all arrival rate vectors $\boldsymbol{\lambda}$ in the closure of Λ, but not in the interior of Λ.

(b) Delay. We define the system delay for a given rate schedule by the sum of the delays of all users, i.e., $\sum_{n\in N} \mathbb{E}[Q^n(t)]$ (assuming its existence), which naturally relates to delay from Little's law [21] in standard queueing theory.

(c) Power-Efficiency. Power consumption in networking and communication systems is receiving an increasing amount of attention due to the recent interest in green information and communication technology. In this paper, we define the long-term averaged transmit power as the power-efficiency of a rate schedule, i.e., $\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \sum_{n\in N} \sum_{k\in K} s_k^n(t)$.

(d) Complexity. Scheduling algorithms with different complexities typically achieve different performances. The notion of complexity in this paper refers to the temporal complexity, which is the computational

complexity (or similarly the execution time) to compute the rate schedule per time slot.[6]

Our goal is now to develop a scheduling algorithm for an N-user DSL system, that achieves throughput-optimality *without* knowledge of the mean arrival rates, where we study the tradeoffs between delay, power-efficiency, and complexity.

III. Throughput-Optimal Scheduling and DSM Algorithms

In this section, we first explain how conventional DSM algorithms can be fit into a cross-layer DSM framework and can be used as building blocks of scheduling algorithms, and then we present throughput-optimal scheduling.

Conventional DSM algorithms aim at optimizing physical layer performance, e.g. allocate transmit powers so as to maximize data rates subject to power constraints, i.e.,

$$\begin{aligned} \max_{\boldsymbol{s}} \quad & \sum_{n\in N} w^n R^n \\ \text{s.t.} \quad & \sum_{k\in K} s_k^n \le P^n, n\in N, \qquad 0 \le s_k^n \le s_k^{n,\text{mask}}, n\in N, k\in K, \end{aligned} \tag{7}$$

where $(w^n : n \in N)$ are user priority weights.

We now describe a scheduling algorithm that is throughput-optimal, and then show its connection to the conventional DSM problem formulation (7). To this end, we first define the *weight* of a rate schedule $\boldsymbol{R} = (R^n : n \in N)$ with respect to a given queue vector $\boldsymbol{Q} = (Q^n : n \in N)$, denoted by $W(\boldsymbol{R}, \boldsymbol{Q})$, as their inner product, i.e.,

$$W(\boldsymbol{R}, \boldsymbol{Q}) \triangleq \sum_{n\in N} Q^n R^n.$$

Consider the following scheduling algorithm, referred to as *Max-Weight (MW)* scheduling: at time slot t, it schedules $\boldsymbol{R}^\star(t)$ that maximizes the weight for the queue length vector at time slot t, i.e.,

$$\boldsymbol{R}^\star(t) \in \arg\max_{\boldsymbol{R}\in\mathcal{R}} W(\boldsymbol{R}, \boldsymbol{Q}(t)),$$

where if there exist multiple max-weight rate schedules, a random tie-breaking is applied.

It has been proved that MW scheduling is throughput-optimal under slightly different system models (e.g., [22]), and extension to our system model is straightforward. By incorporating the DSM physical layer resources, as introduced in section II-A, into the scheduling algorithm, it can be seen that MW scheduling

[6]For distributed algorithms, complexity mostly includes the time to exchange messages among different processing units, whereas in centralized algorithms, complexity is merely the time to finish the corresponding operation in one processing unit (typically measured by the number of CPU operations). We do not explicitly distinguish between distributed and centralized algorithms in this paper, and just use the temporal complexity as the complexity measure.

comes down to solving the following optimization problem at *every* time slot t:

$$\begin{aligned} \max_{\boldsymbol{s}} \quad & \sum_{n\in N} Q^n(t) R^n \\ \text{s.t.} \quad & \sum_{k\in K} s_k^n \leq P^n, n \in N, \qquad 0 \leq s_k^n \leq s_k^{n,\text{mask}}, \quad n \in N, k \in K. \end{aligned} \tag{8}$$

MW scheduling within the DSM setting can thus be considered as solving problem (8) for each time slot t. Problem (8) exactly corresponds to the conventional DSM problem (7), where the weights[7] are equal to the queue lengths, i.e., $w_n = Q^n(t)$, for each time slot t. This means that DSM algorithms developed to solve (7) can be reused as a building block of MW scheduling, to solve problem (8) for each time slot t.

Taxonomy of DSM Algorithms

At this point we would like to distinguish between three different types of DSM algorithms:

(i) *Globally optimal DSM algorithms.* These algorithms succeed in finding the globally optimal solution of (7) and thus also (8) [8]. Examples include optimal spectrum balancing (OSB) [9], branch-and-bound optimal spectrum balancing (BB-OSB) [10] and prismatic branch-and-bound spectrum balancing (PBB) [11]. Note that global optimality is typically defined in a discrete setting, as mentioned in [9]. This means that the globally optimal solution corresponds to the data rate allocations from the discretized rate region as follows

$$\boldsymbol{R}^\star(t) \quad \in \quad \arg\max_{\boldsymbol{R}\in\hat{\mathcal{R}}} W(\boldsymbol{R}, \boldsymbol{Q}(t)), \tag{9}$$

which results in a solution with discretized transmit powers. We want to emphasize here that we will use this typical discrete definition of globally optimal DSM in the remainder of the paper.

(ii) *Locally optimal DSM algorithms.* These algorithms only guarantee a locally optimal solution to (7) and (8). Depending on the initial point, locally optimal algorithms may converge to the globally optimal solution [7]. Examples include distributed spectrum balancing (DSB) [7] and modified iterative waterfilling (MIW) [8]. Note that these algorithms are non-discrete, i.e., they result in continuous transmit powers.

(iii) *Heuristic DSM algorithms.* These algorithms do not necessarily ensure globally or locally optimal solutions to (7) and (8). However, some of these algorithms have practically strong merits in near-optimality with reasonably low complexity. Examples include iterative waterfilling (IW) [5], autonomous spectrum balancing (ASB) [6], and autonomous spectrum balancing 2 (ASB2) [7].

MW scheduling based on a globally optimal DSM algorithm is throughput-optimal. However, globally optimal DSM algorithms have an intractable computational complexity (i.e., exponential in the number of

[7]This weight differs from the weight of a "rate schedule", but for simplicity we use the same term for both cases.

[8]This definition takes the standard assumption [9]–[11] into account that the number of tones is large, which is a reasonable assumption for practical systems, so that also strong duality can be asssumed.

users N, as optimization problems (7) and (8) are NP-hard), hence do not allow a practical implementation. Also, MW scheduling maximizes the throughput-region without explicitly considering the power-efficiency over time. Sections IV and V will address these two issues, respectively.

IV. Throughput-optimal Scheduling with Polynomial Time Complexity

In contrast to globally optimal DSM algorithms, locally optimal DSM algorithms only require polynomial time complexity [7]. However, as the nonconvex problem (7) can have many local optima, locally optimal DSM algorithms sometimes fail to find the globally optimal solution. This strongly depends on the chosen *initial point*. Some initial points lead to a globally optimal solution whereas others lead to a locally optimal solution which can correspond to a quite suboptimal performance [7]. Locally optimal DSM algorithms also have the property of being monotonically increasing. We will exploit the dependence on the initial point, the monotonically increasing property and the typical discrete definition of global optimality, in conjunction with an appropriate scheduling to design polynomial complexity throughput-optimal scheduling algorithms.

A. δ-Randomized DSM Algorithms and Random Rate Scheduling (δ):

To achieve our goal, we first introduce the notion of δ-randomized DSM algorithms, as follows:

Definition 4.1 (δ-randomized DSM algorithm): A DSM algorithm, which produces a random rate schedule $\boldsymbol{R}'(t)$, is δ-randomized for some $0 < \delta \leq 1$, if at each time slot t,

$$\mathbb{P}[\boldsymbol{R}'(t) \geq \boldsymbol{R}^{\star}(t) \mid \boldsymbol{Q}(t)] \geq \delta, \tag{10}$$

where $\boldsymbol{R}^{\star}(t)$ is the globally optimal (discrete) solution of (9).

In other words, a δ-randomized DSM algorithm randomly generates a rate schedule $\boldsymbol{R}'(t)$ that is guaranteed to be at least as good as the optimal (discrete) rate schedule $\boldsymbol{R}^{\star}(t)$ with positive probability δ. Note that we use an inequality in (10), i.e., $\boldsymbol{R}'(t) \geq \boldsymbol{R}^{\star}(t)$, because DSM algorithms may generate a continuous solution for $\boldsymbol{R}'(t)$, whereas the optimal rate schedule $\boldsymbol{R}^{\star}(t)$ is defined for discrete rates. Clearly, a globally optimal DSM algorithm is a 1-randomized DSM algorithm.

Now, consider the following scheduling algorithm using a δ-randomized DSM algorithm, referred to as *Random Rate Scheduling (δ)* or *RRS(δ)*:

Algorithm 1 Random Rate Scheduling(δ): at time slot t

Step 1 Select a random rate schedule $\boldsymbol{R}'(t)$ by solving (8) using a δ-randomized DSM algorithm.

Step 2 Compute the weight of $\boldsymbol{R}'(t)$, i.e., $W(\boldsymbol{R}'(t), \boldsymbol{Q}(t))$.

Step 3 Compare $W(\boldsymbol{R}'(t), \boldsymbol{Q}(t))$ and $W(\boldsymbol{R}(t-1), \boldsymbol{Q}(t))$, and select the rate schedule with larger weight as the rate schedule at time slot t, i.e., $\boldsymbol{R}(t) = \arg\max_{\boldsymbol{S} \in \{\boldsymbol{R}'(t), \boldsymbol{R}(t-1)\}} W(\boldsymbol{S}, \boldsymbol{Q}(t))$.

RRS(δ) can be interpreted as a randomized scheduling that produces a "reasonably good" rate schedule in terms of its non-zero probability of finding a globally optimal rate schedule (**Step 1**), in conjunction with progressively selecting better rate schedules by comparing the previous rate schedule and the current randomized rate schedule (**Step 3**). Again, when $\delta = 1$, RRS(δ) corresponds to MW scheduling. Theorem 4.1 states the throughput property of the RRS(δ).

Theorem 4.1: For any $0 < \delta \leq 1$, RRS(δ) with a δ-randomized DSM algorithm is throughput-optimal.

A similar result has been proved under different systems such as switching systems (e.g., see the seminal work [23]) or wireless networks (e.g., see [24]). The proof is presented in Appendix for completeness. The key intuition in Theorem 4.1 is that the throughput is determined by stability of the system that is measured over a long-term period. The maximum stability is guaranteed by, rather than applying a MW rate schedule in every slot, infrequently applying MW rate schedules at some time slots, together with using suboptimal, yet "reasonably good" rate schedules elsewhere. In Algorithm 1, infrequent MW rate schedules are realized by probabilistic selections and "reasonably good" rate schedules are chosen by selection of a rate schedule with larger weights between a random rate schedule and the previous rate schedule.

The δ-randomized DSM algorithms thus lead to a parameterized family of scheduling algorithms that achieve throughput-optimality. The parameter δ can range from 1 to a very small number, where typically, a smaller δ corresponds to scheduling algorithms with lower complexity at the cost of increasing delays, as will be discussed in Sections IV-B and IV-C.

The remaining task is to develop δ-randomized DSM algorithms that have only polynomial time complexity and are thus much more practically feasible than globally optimal DSM algorithms with exponential time complexity. For this we consider a locally optimal DSM algorithm and extend it with a *random selection of the initial point for the transmit powers* $\boldsymbol{s} = (s_k^n : k \in K, n \in N)$. More specifically, we define the set of all feasible elements, i.e., $\mathcal{D}$, as follows

$$\mathcal{D} = \{(y_k^n : k \in K, n \in N) \mid \forall n \in N, k \in K : y_k^n \in \mathcal{D}_k^n, \forall n \in N : \sum_{k \in K} y_k^n \leq P^n\}.$$

By choosing a point of set $\mathcal{D}$ with a uniform probability distribution, we have a non-zero possibility for each point of the set to be chosen. This point is now chosen as an initial point for the locally optimal DSM algorithm. As a locally optimal DSM algorithm [7] is monotonically increasing over its successive iterations, it converges to a (locally optimal) point that is at least as good as this initial point. As there is also a non-zero probability to choose the globally optimal (discrete) solution, we have a non-zero probability to converge to a solution that is at least as good as this discrete optimum. Combining a locally optimal DSM algorithm with a random initial point taken from the feasible discrete set with a uniform probability distribution, thus results in a DSM algorithm that produces a globally optimal solution to (8) with non-zero probability. This satisfies the definition of a δ-randomized DSM algorithm.

Our first δ-randomized DSM algorithm will be referred to as **R1-DSM**, and is defined as follows

---

R1-DSM (Random Initial Point)

1) Pick a random initial transmit power $\boldsymbol{x} \in \mathcal{D}$ with uniform probability distribution,
2) Apply a locally optimal DSM algorithm with $\boldsymbol{x}$ as the initial point.

Taking Theorem 4.1 into account, we can see that **R1-DSM** in combination with the RRS(δ) (Algorithm 1) results in a throughput-optimal DSM scheduling scheme by the usage of polynomial time complexity algorithms, for which the (theoretical) average lower-bound on δ is $1/|\mathcal{D}|$, with $|\mathcal{D}|$ the cardinality of set $\mathcal{D}$.

Note that $1/|\mathcal{D}|$ is only the probability that we select an initial transmit power which is globally optimal. By running locally optimal algorithms, our actual δ may become much larger, since δ should roughly be:

$$\delta \approx \frac{\text{the number of initial points that provide convergence to the globally optimal solution}}{|\mathcal{D}|}.$$

For small interference cases, problem (8) is convex [25], and thus $\delta = 1$. However for general cases this is not true and δ can be much smaller.

B. Delay Performance of RRS(δ)

As shown in the previous section, it is possible to achieve throughput-optimality with RRS(δ) and a randomized locally optimal DSM algorithm. In this section, we show that the price paid for the reduction from exponential to polynomial complexity without losing throughput, is *delay*.

Calculating the exact delay performance in our system is known to be very difficult, mainly due to the complex coupling of queueing dynamics across users, tones, and stochastic arrivals. Therefore we rely on a delay bound. Although this delay bound may not be tight in some scenarios, it is quite helpful to understand how delay performance scales with δ, which in turn relates to the computational complexity of scheduling algorithms.

The delay performance should depend on the arrival rate vector $\boldsymbol{\lambda}$, e.g., when $\boldsymbol{\lambda}$ approaches the boundary of the throughput-region, then delay will correspondingly increase. To quantify this intuition, we use the notion of distance between the arrival rate vector and the boundary of the throughput-region as follows:

Definition 4.2 (Distance):

$$d(\boldsymbol{\lambda}) = \sup\{\epsilon : \boldsymbol{\lambda} \in (1-\epsilon)\Lambda^b\}. \tag{11}$$

This distance essentially represents how heavily the system is loaded, where a smaller $\boldsymbol{\lambda}$ leads to a larger $d(\boldsymbol{\lambda})$. We note that the distance cannot take the value of 0 for arrival rate vectors in the throughput-region, which is defined as an open set, and taking Definitions 2.2 and 2.3 into account. Using Definition 4.2, the following bound on the system delay can be obtained for randomized scheduling with δ-randomized DSM algorithm and $\boldsymbol{\lambda} \in \Lambda$:

Theorem 4.2:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{n\in N}\mathbb{E}\Big[Q^n(t)\Big] \;\leq\; \frac{N^2\tilde{R}\Omega_2}{2d(\boldsymbol{\lambda})} + \frac{2\Omega_1 N\tilde{R}}{\delta d(\boldsymbol{\lambda})}, \tag{12}$$

where $\tilde{R}$ is a smallest constant such that $R_{\max}\tilde{R} \geq 1$ if $R_{\max} < 1$, and $\tilde{R} = 1$ otherwise. Note that Ω_1 and Ω_2 are defined in Section II-A. The proof is presented in the Appendix.

Theorem 4.2 shows the tradeoff between complexity and delay, where we observe that as δ decreases, the delay bound increases. The δ refers to the probability that RRS(δ) finds a globally optimal rate schedule, where a larger δ typically requires more searching and smart rate scheduling, and thus more complexity. For RRS(δ) with **R1-DSM**, as discussed earlier, $\delta \geq \frac{1}{|\mathcal{D}|}$ on average, and so the delay bound may become very large, i.e., delay bound $\approx O(|\mathcal{D}|)$. Thus, RRS(δ) with **R1-DSM** is more suitable for elastic data applications, but may be inappropriate for real-time voice or video. This motivates us to develop delay-enhanced algorithms, which will be presented in the next subsection. However, note that although the delay bound may become very large, it will always remain finite and thus R1-DSM is throughput optimal. This is a consequence of the typical discrete definition of global optimality in DSL DSM that we exploit.

C. Delay-Enhanced RRS(δ)

Locally optimal DSM algorithms reduce complexity, which guarantees finding the globally optimal solution only probabilistically (in conjunction with a random initial point) with a negative impact on the delay performance when used with RRS(δ). The question arises if we can improve this delay performance significantly, by exploiting the specific structure of our problem. The answer is positive, and we will use the following problem-specific temporal-spectral correlation (TSC) property to obtain delay-enhanced RRS(δ) algorithms:

- ***Temporal correlation.*** When queue lengths are large (i.e., the arrival rate vector is close to the boundary of the throughput-region), the system does not observe much difference in weights (i.e., queue lengths) over subsequent time slots.
- ***Spectral correlation.*** Subsequent tones have similar channel characteristics and so also have similar optimal transmit powers.

This problem-specific TSC property can actually be exploited to choose the initial points more efficiently. One can add two additional initial points to the random initial point for each tone $k \in K$: (1) the best local optimum at tone k from the previous time slot using the idea of *temporal correlation*, and (2) the best local optimum from tone $k-1$ using the idea of *spectral correlation*. The addition of these extra initial points will increase δ of the randomized DSM algorithms, resulting in better delay performance of RRS(δ). Hence, the following δ-randomized DSM algorithm, referred to as R2-DSM, can be combined with RRS(δ) (Algorithm

1) to obtain a throughput-optimal scheduling with significantly improved delay performance, compared to RRS(δ) with **R1-DSM**.

R2-DSM (Temporal-Spectral Correlation)

1) For each tone $k \in K$, choose three initial points: *i)* random point $\boldsymbol{x} \in \mathcal{D}_k$ with uniform probability distribution, *ii)* the best local optimum at tone k from the previous time slot, *iii)* the best local optimum from tone $k-1$,
2) Apply a locally optimal algorithm for all initial points and retain best local optimum in each tone.

A third δ-randomized DSM algorithm is inspired by the recent result in [26], where it is stated that for large crosstalk scenarios the solution of (7) is an FDMA solution, i.e., a solution where only one user is active at each tone. It is indeed observed that for tones with large crosstalk the locally optimal solutions get isolated along the axes [7]. Therefore in addition to the random initial point, we propose to extend the number of initial points in each tone $k \in K$ so that it includes all the solutions where only one user transmits at spectral mask and the transmit powers for all other users are set to zero. By adding these initial points, the likelihood that one of these initial points leads to the globally optimal solution, increases significantly. This results in the following δ-randomized DSM algorithm, referred to as R3-DSM:

R3-DSM (N single-user initial points)

1) For each tone $k \in K$, choose $N+1$ initial points: *i)* random point $\boldsymbol{x} \in \mathcal{D}_k$ with uniform probability distribution, *ii)* N initial points with each having only one (different) user active at spectral mask,
2) Apply locally optimal algorithm for all initial points and retain best local optimum in each tone.

Note that RRS(δ) with **R1-DSM** as well as with **R2-DSM**, and **R3-DSM** is provably throughput-optimal. In terms of complexity, **R2-DSM** requires 2 times more complexity than **R1-DSM**, and **R3-DSM** requires N times more complexity than **R1-DSM**. In terms of δ, which determines the delay performance, we have that $\delta_{\mathrm{R1-DSM}} \leq \delta_{\mathrm{R2-DSM}} \simeq \delta_{\mathrm{R3-DSM}}$. In fact for DSL scenarios for which (8) has many locally optimal solutions, we observe that $\delta_{\mathrm{R1-DSM}} \ll \delta_{\mathrm{R2-DSM}} \simeq \delta_{\mathrm{R3-DSM}} \simeq 1$ which means that the delay performance is much better for RRS(δ) with **R2-DSM** and **R3-DSM** for just a slight increase in complexity. This will be verified in Section VI.

V. Power-Efficient Throughput-optimal Scheduling

We have so far discussed throughput-optimal scheduling based on the MW rule, proposed a polynomial time complexity throughput-optimal scheduling exploiting locally-optimal DSM, and studied the impact of complexity on delay performance. A disadvantage of these throughput-optimal scheduling algorithms is that

the underlying δ-randomized DSM algorithms that solve (8), always result in a transmit power allocation that corresponds to some point on the boundary of the achievable rate region $\mathcal{R}$. These boundary points typically correspond to operating points that consume full power, and are thus not very power-efficient. We now study a scheduling algorithm that additionally incorporates power minimization, and yet is still provably throughput-optimal.

A. GMW (Green-Max-Weight) Scheduling

We first propose a variant of MW scheduling, referred to as Green-Max-Weight (GMW) scheduling, for which we define the *greening-weight* $V(\boldsymbol{R}, \boldsymbol{Q}, \beta)$ of a rate schedule $\boldsymbol{R}$ with respect to a queue length vector $\boldsymbol{Q}$ and parameter β as:

$$V(\boldsymbol{R}, \boldsymbol{Q}, \beta) \triangleq \sum_{n \in N} Q^n R^n - \beta S(\boldsymbol{R}),$$

where $S(\boldsymbol{R})$ is the *total sum of powers* over all users n and tones k corresponding to the rate vector $\boldsymbol{R}$.

Then, GMW scheduling is described as follows: at time slot t, it schedules $\boldsymbol{R}_G^\star(t)$ that maximizes the greening-weight for the queue length vector $\boldsymbol{Q}(t)$ at time slot t, i.e.,

$$\boldsymbol{R}_G^\star(t) \in \arg\max_{\boldsymbol{R} \in \mathcal{R}} V(\boldsymbol{R}, \boldsymbol{Q}(t), \beta),$$

which is expressed by the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{s}} \quad & \sum_{n=1}^{N} Q^n(t) R^n - \beta \sum_{n=1}^{N} \sum_{k=1}^{K} s_k^n \\ \text{s.t.} \quad & \sum_{k=1}^{K} s_k^n \leq P^n, n \in N, \qquad 0 \leq s_k^n \leq s_k^{n,\text{mask}}, \ n \in N, k \in K, \end{aligned} \tag{13}$$

with β being a constant.

As can be seen from (13), GMW scheduling selects rate schedules that maximize a weighted sum of aggregate data rates and the negative aggregate transmit power consumptions. It is easy to see that GMW scheduling is also computationally intractable, i.e., NP-hard, similar to MW scheduling. In this section, we will apply an analogous randomized idea to reduce complexity at the cost of increasing delay. Two fundamental questions should be answered: *(i)* Is GMW scheduling throughput-optimal? and *(ii)* What is the cost of considering power consumption in selecting a rate schedule? We will also address these two questions in the next section.

B. Green Random Rate Scheduling(δ)

We consider a scheduling algorithm, referred to as *Green Random Rate Scheduling (δ)* or *GRRS(δ)*. GRRS(δ) is a randomized variant of GMW scheduling which reduces complexity, similarly to the case of MW scheduling and RRS(δ). We let $V(\boldsymbol{R}(t), \boldsymbol{Q}(t), \beta)$ be the greening-weight of a rate schedule that GRRS(δ) chooses at time slot t.

Algorithm 2 Green Random Rate Scheduling(δ): at time slot t

Step 1 Select a random rate schedule $\boldsymbol{R}'(t)$ by solving (13) using a δ-randomized DSM algorithm.

Step 2 Compute the weight of $\boldsymbol{R}'(t)$, i.e., $V(\boldsymbol{R}'(t), \boldsymbol{Q}(t), \beta)$.

Step 3 Compare $V(\boldsymbol{R}'(t), \boldsymbol{Q}(t), \beta)$ and $V(\boldsymbol{R}(t-1), \boldsymbol{Q}(t), \beta)$, and select the rate schedule with larger weight as the rate schedule at time slot t, i.e., $\boldsymbol{R}(t) = \arg\max_{\boldsymbol{S} \in \{\boldsymbol{R}'(t), \boldsymbol{R}(t-1)\}} V(\boldsymbol{S}, \boldsymbol{Q}(t), \beta)$.

Theorem 5.1 specifies the throughput-optimality and delay performance of GRRS(δ).

Theorem 5.1: GRRS(δ) is throughput-optimal and an upper-bound on the delay performance of GRRS(δ) is given by:

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{n \in N} \mathbb{E}\Big[Q^n(t)\Big] \quad \leq \quad \frac{N^2 \tilde{R} \Omega_2}{2d(\boldsymbol{\lambda})} + \frac{2\Omega_1 N \tilde{R}}{\delta d(\boldsymbol{\lambda})} + \frac{\beta N \tilde{R} (\sum_n P^n)}{d(\boldsymbol{\lambda})}, \tag{14}$$

where note that $\sum_n P^n$ is the total power budget across users.

The proof is presented in the Appendix. From Theorem 5.1, we derive the following statements: *(i)* GMW is throughput-optimal, because GMW corresponds to GRRS(δ) with $\delta = 1$; *(ii)* the cost paid to reduce power consumption while preserving throughput-optimality is again increased delay, which linearly scales with the parameter β. Note that β corresponds to the weighting that is given to power consumption minimization with respect to data rate maximization. Thus, as β increases, we can achieve throughput-optimality with less power, but then delay also increases, where the right-most term of (14) is the additional delay by considering the power consumption.

In Algorithm 2, slightly modified DSM algorithms are needed to solve (13). In [12], [13], DSM algorithms are presented that can solve such modified problem formulations. By combining these DSM algorithms with one or more initial points (exploiting e.g. the TSC property) as in Sections IV-B and IV-C, we obtain similar δ-randomized DSM algorithms that have property (10) with $\boldsymbol{R}^\star(t)$ replaced by $\boldsymbol{R}^\star_G(t)$, and with global optimality defined in a discrete manner similarly as in (9). In this way, we obtain similar algorithms as R1-DSM, R2-DSM and R3-DSM that trade-off delay and computational complexity. An increasing value for β furthermore increases delay but improves the average power-efficiency of the DSL system.

VI. Simulation Results

A. Simulation Setup and Mixed Deployment Scenarios

The following system parameters are used for the simulations: The twisted pair lines have a diameter of 0.5 mm (24 AWG). The SNR gap Γ is 12.9 dB, corresponding to a coding gain of 3 dB, a noise margin of 6 dB and a target symbol error probability of 10^{-7}. The tone spacing Δ_f is 4.3125 kHz. The DMT symbol rate f_s is 4 kHz. The maximum per-user transmit power is 20.4 dBm. In current standards, the discrete

granularity is set to $\Delta = 0.5$ dBm/Hz [27]. We consider the following two mixed deployment topological scenarios, i.e., including central office (CO) lines as well as remote terminal (RT) lines:

1) A *two-user* ADSL downstream scenario with line lengths 5000m and 3000m. Line 1 starts from the CO and is also referred to as the CO-user. Line 2 starts from a RT which is located 3000m from the CO, and is also referred to as the RT-user. This is a standard near-far scenario, which is known as a very challenging scenario for which locally optimal DSM algorithms generally fail to converge to the globally optimal solution [7].
2) A *six-user* ADSL downstream scenario with line lengths 5000m, 4000m, 3500m, 3000m, 3000m, and 2500m. Lines 1 and 2 start from the CO. Lines 3 and 4 start from a first remote terminal (RT1) which is located 500m from the CO. Lines 5 and 6 start from a second remote terminal (RT2) which is located 3000m from the CO. This scenario is depicted in Figure 1.

In Section VI-E we will furthermore comment how the obtained simulation results extend for non-mixed deployment scenarios.

B. Throughput-Optimality of Random Rate Scheduling(δ)

In Figure 2 we show the two-user throughput-regions for the random rate scheduling algorithms in Section IV, each of which uses a different DSM algorithm to solve **Step 1** of Algorithm 1. We used locally optimal DSM algorithms (DSB, MIW) with different sets of fixed (i.e., non-random) initial points, with 'zeros' referring to initial zero transmit powers, and 'mask' referring to initial transmit powers at spectral mask. Note that the throughput of RRS(δ) with the δ-randomized DSM algorithms (**R1-DSM**, **R2-DSM**, **R3-DSM**) is optimal as opposed to RRS(δ) with the locally optimal DSM algorithms with fixed initial points, whose throughput-regions are sub-optimal and depend on the initial points. In Figure 3 we show the delay performance of RRS(δ) combined with DSB (or similarly for MIW) with fixed zeros as initial points and combined with **R2-DSM**, for the arrival rate vector indicated with the cross in Figure 2. One can see that RRS(δ) with **R2-DSM**, which exploits the TSC property, succeeds in stabilizing the system with only small delay, while this is not the case for fixed initial points and after 90000 time slots.

C. Impact of δ for δ-Randomized DSM Algorithms

In Figure 4 the per-tone probabilities are shown for the 2-user case for RRS(δ) scheduling combined with the proposed δ-randomized algorithms (using DSB). More specifically, for each tone, the per-tone probability gives the probability that the considered locally optimal DSM algorithm finds the optimal solution in that tone, with initial points chosen from $\mathcal{D}$ with uniform probability distribution and with discrete granularity $\Delta = 0.5$ dBm/Hz. It can be seen that **R1-DSM** has small probabilities in a subset of 30 tones. This results in an overall very small probability $\delta = 4.5 \times 10^{-23}$, which is obtained by multiplying the probabilities over all tones. However, **R2-DSM** and **R3-DSM** have larger probabilities over all tones, resulting in an

overall probability of $\delta = 0.9388$ and $\delta = 1$, respectively. The small additional computational complexity of **R2-DSM** and **R3-DSM** thus results in a huge increase of δ and thus also an improved delay performance. We see a typical trend in our simulations that RRS(δ) with **R3-DSM** has a much better delay performance compared to RRS(δ) with **R1-DSM**, where both of them are throughput-optimal.

D. Green Random Rate Scheduling(δ)

In Figure 5 we plot the time averaged transmit powers, time averaged data rates and the delay performance for increasing β (from left to right) for GRRS(δ) with **R1-DSM** and using DSB as a locally optimal DSM algorithm, and for the 2-user ADSL scenario. The input arrival rate vector corresponds to a CO rate of 0.85 Mbps and an RT rate of 5.95 Mbps. A larger value for β clearly results in less power consumption over time but also a worse delay performance. Note that the variation in data rate is very small. A similar observation has already been made in [12], where it was shown that large transmit power savings can be achieved with only a minor degradation in data rate performance.

It can be seen that for this scenario a good trade-off between data rates, transmit powers and delay performance is obtained by choosing $\beta = 10^6$. This choice achieves a large part of the transmit power saving while maintaining delay performance, and without impacting the data rates too much. A larger value for β results in a much worse delay performance. This demonstrates that the β parameter indeed offers the possibility to trade-off average data rates, transmit powers and delay.

Figure 6 shows the convergence behaviour and variation of the data rates, total transmit powers and queue lengths for the six-user ADSL scenario. The mean of the arrival rate vector corresponds to 90% of the data rates of the unweighted rate region boundary point. The GRRS(δ) algorithm with **R1-DSM** (and DSB) is used to dynamically update the data rates and transmit powers. Both data rates and transmit powers vary so as to keep the queues stable. It can be seen that the queues are stabilized after 5000 time slots. Note that significant power savings are obtained, i.e., the power usage is well below the 100% full power usage.

Figure 7 shows the cumulative probability distributions (CDF) of the data rates, total transmit powers and queue lengths for the six-user ADSL scenario in steady state operation, i.e., when the queues are stabilized. Modem 1 (in blue) corresponds to a long line, is a small interferer and is subject to strong interference of the other lines. It has a small variation in steady state operation, whereas the other lines, which are large interferers, have a larger variation. This is because large interferers have to balance between protecting the weak user (Modem 1) and protecting themselves, resulting in larger resource variations in time.

E. Non-Mixed Deployment Scenarios

The above simulations considered mixed deployment scenarios, in which the crosstalk impact can be quite asymmetric, i.e., some lines cause much more interference to others. It is shown in [7] that for those scenarios, locally optimal DSM algorithms with one fixed initial point may result in very suboptimal data

rate performance. The extension with random initial points into the proposed random rate scheduling scheme combined with δ-randomized DSM algorithms then results in a significant improved throughput and delay performance, and power-efficiency as shown in Sections VI-B, VI-C, and VI-D. For CO-only scenarios, i.e., non-mixed scenarios, with similar (long) line lengths it is shown in [7] that locally optimal DSM algorithms (with fixed initial points) typically perform near-optimal and consequently the corresponding throughput and delay performance, and power-efficiency, will be similar compared to δ-randomized DSM algorithms. However, as the number of asymmetric users in the considered cable bundle increases, the performance of existing locally optimal DSM algorithms becomes much worse compared to the proposed schemes.

VII. Conclusion

It is crucial to understand and design algorithms for DSL systems from the perspective of quality-of-service as experienced by the users. While existing literature on DSL DSM has focused on physical layer performance metrics, such as data rates and transmit powers, we have extended this approach towards a cross-layer (time-dynamic) framework that allows to study throughput and delay performance, which are crucial higher layer performance metrics for many real-time applications. More specifically, by jointly considering scheduling and physical layer DSM, we have showed that even sub-optimal DSM algorithms can achieve throughput-optimality at the cost of an increased system delay. We have also proposed extensions to significantly reduce delay performance with small additional complexity, which make our proposed schemes more practical in real systems. The proposed solutions exploit the specific structure of DSL DSM problems and algorithms. Finally, we have proposed a second framework that also focuses on reducing the consumed transmit power, yet sustain throughput-optimality, but again at the cost of an increased delay.

VIII. Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions which have improved the quality and presentation of the paper.

References

[1] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen, "Throughput and delay of DSL dynamic spectrum management with dynamic arrivals," in *Proc. of IEEE GLOBECOM*, New Orleans, LA, USA, Nov. 2008, pp. 1–5.

[2] F. Vanier, "World broadband statistics: Q4 2010," Point Topic Ltd, http://www.point-topic.com, Tech. Rep., Mar. 2011.

[3] K. B. Song, S. T. Chung, G. Ginis, and J. M. Cioffi, "Dynamic spectrum management for next-generation DSL systems," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 101–109, Oct. 2002.

[4] G. Ginis and J. M. Cioffi, "Vectored transmission for digital subscriber line systems," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1085–1104, Jun. 2002.

[5] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1105–1115, Jun. 2002.

[6] R. Cendrillon, J. Huang, M. Chiang, and M. Moonen, "Autonomous spectrum balancing for digital subscriber lines," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4241–4257, August 2007.

[7] P. Tsiaflakis, M. Diehl, and M. Moonen, "Distributed spectrum management algorithms for multiuser DSL networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4825–4843, Oct. 2008.

[8] W. Yu, "Multiuser water-filling in the presence of crosstalk," in *Proc. of Information Theory and Applications (ITA)*, San Diego, CA, USA, 2007.

[9] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multiuser spectrum balancing for digital subscriber lines," *IEEE Transactions on Communications*, vol. 54, no. 5, pp. 922–933, May 2006.

[10] P. Tsiaflakis, J. Vangorp, M. Moonen, and J. Verlinden, "A low complexity optimal spectrum balancing algorithm for digital subscriber lines," *Elsevier Signal Processing*, vol. 87, no. 7, pp. 1735–1753, Jul. 2007.

[11] Y. Xu, T. Le-Ngoc, and S. Panigrahi, "Global concave minimization for optimal spectrum balancing in multi-user DSL networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 4825–4843, Jul. 2008.

[12] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen, "Green DSL: Energy-Efficient DSM," in *Proc. of IEEE ICC*, Dresden, Germany, Jun. 2009.

[13] ——, "Fair greening of broadband access: Spectrum management for energy-efficient DSL networks," *EURASIP Journal on Wireless Communications and Networking*, no. 2011:140, pp. 1–15, Oct. 2011.

[14] J. M. Cioffi, S. Jagannathan, W. Lee, H. Zou, A. Chowdhery, W. Rhee, G. Ginis, and P. Silverman, "Greener copper with dynamic spectrum management," in *Proc. of AccessNets*, Las Vegas, NV, USA, Oct. 2008.

[15] J. M. Cioffi et al., "Vectored DSLs with DSM: the road to ubiquitous gigabit DSLs," in *Proc. World Telecommunications Congress*, Budapest, Hungary, 2006.

[16] K. Seong, R. Narasimhan, and J. M. Cioffi, "Queue proportional scheduling via geometric programming in fading broadcast channels," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1593 –1602, 2006.

[17] B. Li, P. Tsiaflakis, M. Moonen, J. Maes, and M. Guenach, "Dynamic resource allocation based partial crosstalk cancellation in DSL networks," in *Proc. of IEEE GLOBECOM*, Miami, FL, USA, 2010.

[18] T. Starr, J. M. Cioffi, P. J. Silverman, *Understanding digital subscriber lines*. Prentice Hall, 1999.

[19] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Transactions on Communications*, vol. 54, no. 7, Jul. 2006.

[20] Z. Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal on Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[21] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. New York: Wiley–Interscience, 1975.

[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, December 1992.

[23] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology." *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1067–1073, 1997.

[24] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proc. of ACM Sigmetrics*, San Diego, CA, USA, 2007.

[25] P. Tsiaflakis, C. Tan, Y. Yi, M. Chiang, and M. Moonen, "Optimality certificate of dynamic spectrum management in multi-carrier interference channels," in *Proc. of IEEE International Symposium on Information Theory*, Toronto, Jul. 2008.

[26] S. Hayashi and Z.-Q. Luo, "Spectrum management for interference-limited multiuser communication systems," *IEEE Transactions on Information Theory*, vol. 55, no. 3, Mar. 2009.

[27] ITU Std., G.997.1, "Physical layer management for digital subscriber line (DSL) transcievers," 2003.

[28] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. London, UK: Springer-Verlag, 1993.

APPENDIX

Proof of Theorems 4.1 and 4.2: Since the system can be modeled by a Markov chain, the positive recurrent of the chain implies stability, for which we use the Foster's criterion [28]. Consider the quadratic Lyapunov function and the corresponding drift: $L(t) = \sum_n Q^n(t)^2$ and $\Delta L(t) = \mathbb{E}[L(t+1) - L(t)|\boldsymbol{Q}(t)]$. To prove Theorem 4.1, from the Foster's criterion, it suffices to show that for any arrival rate vector $\boldsymbol{\lambda} \in \Lambda$,

there exists a sufficiently large B, such that whenever $\sum_n Q^n(t) \geq B$, $\Delta L(t)$ is negative. Note that if $\boldsymbol{\lambda} \in \Lambda$, we can find $\epsilon > 0$, such that $\boldsymbol{\lambda} \in (1-\epsilon)\Lambda$. Then, we can find $\beta_i \geq 0$, $i = 1, \ldots, |\hat{\mathcal{R}}|$, such that $\sum_i \beta_i < 1$, and $\boldsymbol{\lambda} = (1-\epsilon)\sum_i \beta_i \boldsymbol{R}_i$, where $\boldsymbol{R}_i$ is the i-th rate schedule of the entire valid rate schedule set $\hat{\mathcal{R}}$.

First, we prove that $W(t) \geq W^\star(t) - X$, where X is a random variable with $\mathbb{E}[X] \leq 2\Omega_N/\delta$. Denote by $\{t_i\}_{i=0,1,\ldots}$ the sequence of (random) time slots, where $W(t_i) = W^\star(t_i)$. Let $\Delta t_i = t_{i+1} - t_i$. Now for $\forall t \in [t_i, t_{i+1})$, we should have that (i) $W(t) \geq W(t_i) - \Delta t_i \Omega_1 N$ and (ii) $W^\star(t) \leq W^\star(t_i) + \Delta t_i N \Omega_1$, since the maximum queue length difference at one slot is bounded by Ω_1, where recall that $\Omega_1 = A_{\max} + R_{\max}$. Then, from (i) and (ii), it is easy to prove that $W(t) \geq W^\star(t) - 2\Delta t_i \Omega_1 N$, with $\mathbb{E}[\Delta t_i] \leq 1/\delta$ (because the MW rate schedule is found with at least δ probability). We henceforth let $\xi \triangleq 2\Omega_1 N/\delta$.

Second, we prove that the negative Lyapunov drift is guaranteed for sufficiently large queue lengths. From the queue dynamics in (6) and the fact that if $a \leq [b-c]^+ + d$ then $a^2 \leq b^2 + c^2 + d^2 - 2b(c-d)$, we have $L(t+1) - L(t) = \sum_n (Q^n(t+1)^2 - Q^n(t)^2) = R^n(t)^2 + A^n(t)^2 - 2Q^n(t)(A^n(t) - R^n(t)) \leq R_{\max}^2 + A_{\max}^2 - 2Q^n(t)(A^n(t) - R^n(t))$. Then, we have

$$\begin{aligned}
\Delta L(t) &= \mathbb{E}[L(t+1) - L(t) | \boldsymbol{Q}(t)] \leq 2\sum_n Q^n(t)\left(\mathbb{E}[A^n(t+1)] - R^n(t)\right) + \Omega_2 N \\
&= 2\sum_n Q^n(t)\left(\lambda^n - R^n(t)\right) + \Omega_2 N \leq 2\left((1-\epsilon)\sum_i \beta_i W(\boldsymbol{R}_i) - W(t)\right) + \Omega_2 N \\
&\leq 2\left((1-\epsilon)\sum_i \beta_i W(\boldsymbol{R}_i) - W^\star(t) + W^\star(t) - W(t)\right) + \Omega_2 N \\
&\leq -2\epsilon W^\star(t) + 2\xi + \Omega_2 N \leq -2\epsilon\left(\sum_n Q^n(t)/\tilde{R}N\right) + 2\xi + \Omega_2 N, \qquad (15)
\end{aligned}$$

where (15) follows from $\sum_i \beta_i W(\boldsymbol{R}_i) \leq W^\star(t)$ and (15) follows from the fact that $W^\star(t) \geq R_{\max} \max_{n \in N} Q^n(t)$ (otherwise, we just schedule user n with the rate $R_{\max}$ to have a larger weight) and thus $NW^\star(t) \geq NR_{\max} \max_{n \in N} Q^n(t) \geq \sum_n Q^n(t) R_{\max} \geq \sum_n Q^n(t)/\tilde{R}$. Finally, since ϵ, ξ, and N are fixed, there exists a sufficiently large B, such that whenever $\sum_n Q^n(t) \geq B$, $\Delta L(t)$ is negative, which proves Theorem 4.1.

From (15), $\mathbb{E}\Big[L(j) - L(0)\Big] \leq j(\Omega_2 N + 2\xi) - \frac{2\epsilon}{N\tilde{R}} \sum_{t=0}^{j-1} \sum_n \mathbb{E}[Q^n(t)]$. Then, we deduce:

$$\frac{1}{j}\sum_{t=0}^{j-1}\sum_l \mathbb{E}[Q^n(t)] \leq \frac{N\tilde{R}(\Omega_2 N + 2\xi)}{2\epsilon} - \frac{N\tilde{R}\mathbb{E}[L(j) - L(0)]}{2j\epsilon}. \qquad (16)$$

Theorem 4.2 is proved by letting $j \to \infty$ and $\epsilon = d(\boldsymbol{\lambda})$ to get the tightest upper-bound by the fact that $\xi = 2\Omega_1 N/\delta$. ■

Proof of Theorem 5.1: Let $V^\star(t)$ be the weight of GMW. Then, it is easy to prove that $V(t) \geq V^\star(t) - X \geq W^\star(t) - \beta\sum_n P^n - X$, where X is again a random variable with $\mathbb{E}[X] \leq 2\Omega_1 N/\delta$. The stability can be proved similarly to (15) by regarding ξ as a different number, i.e., $\xi = \beta\sum_n P^n - 2\Omega_1 N/\delta$ and the delay can be computed by the new ξ in (16).

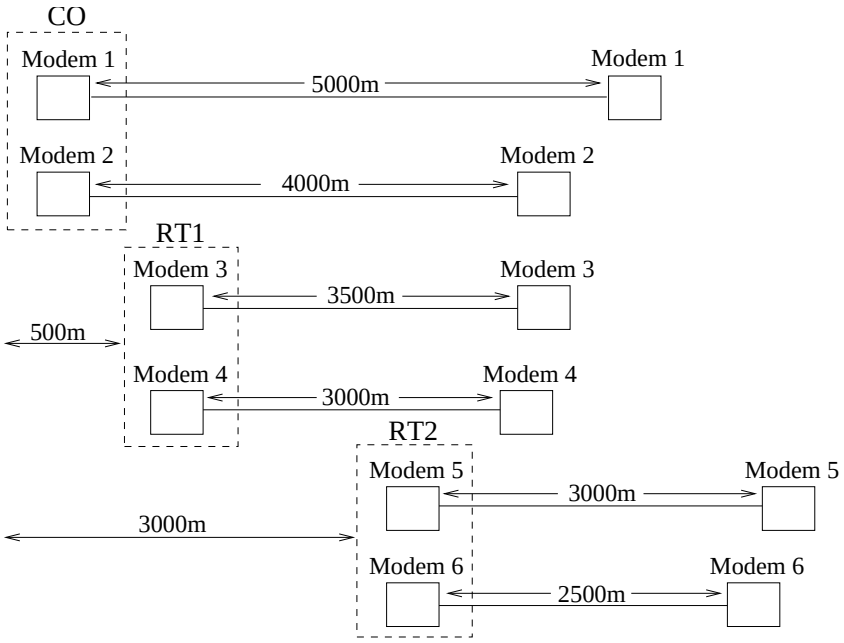


Fig. 1. Six-user ADSL downstream scenario.

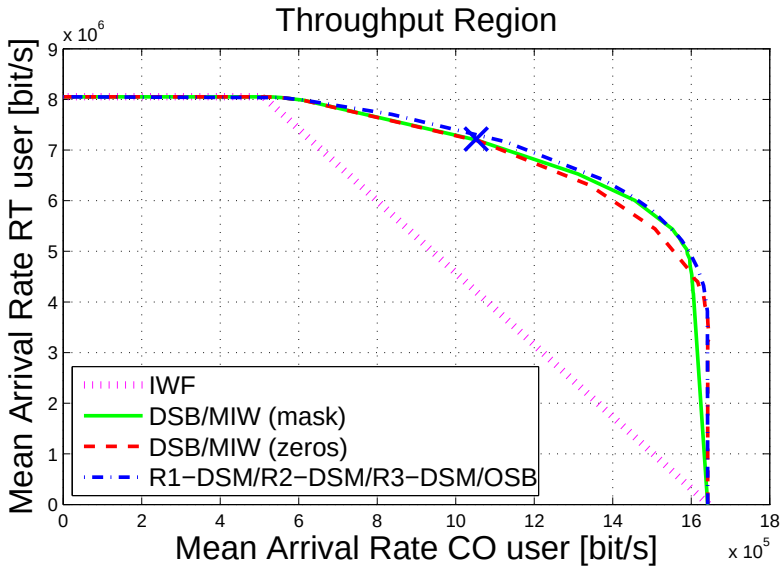


Fig. 2. Throughput-region for 2-user ADSL scenario. The curve indicated by DSB/MIW (mask) refers to the throughput-region of the DSB and MIW algorithms with fixed initial point equal to spectral mask, i.e. $s_k^n = s_k^{n,\text{mask}}, \forall n, k$. The curve indicated with DSB/MIW (zeros) starts from the all zero initial point. The cross indicates the (mean) arrival rate vector used to generate the delay performances for RRS(δ) combined with DSB (with fixed zeros) and for R2-DSM of Figure 3.

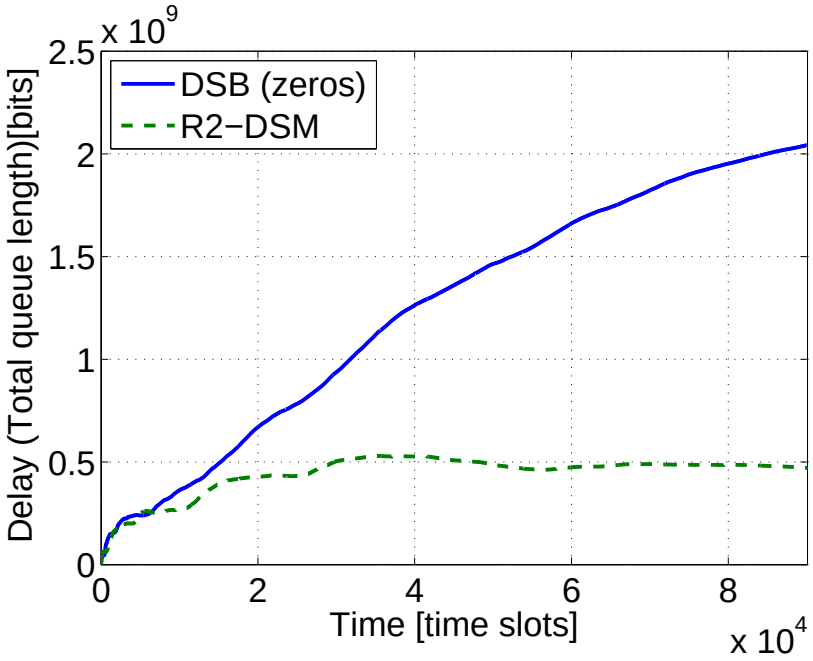


Fig. 3. Delay behaviour of RRS(δ) with DSB algorithm (with fixed zero initial points) and with R2-DSM for 2-user ADSL scenario.

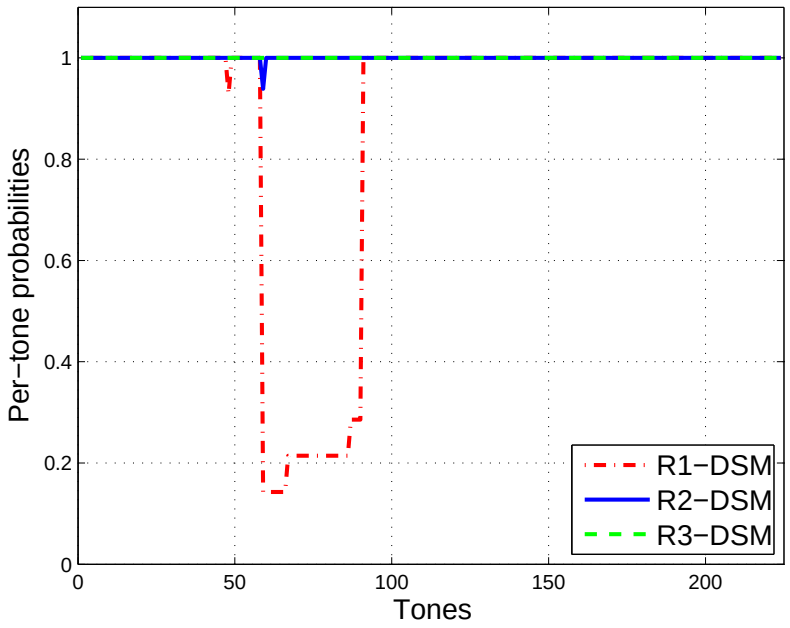


Fig. 4. Per-tone probabilities for convergence to per-tone globally optimal solution for DSB with random initial point for 2-user ADSL scenario.

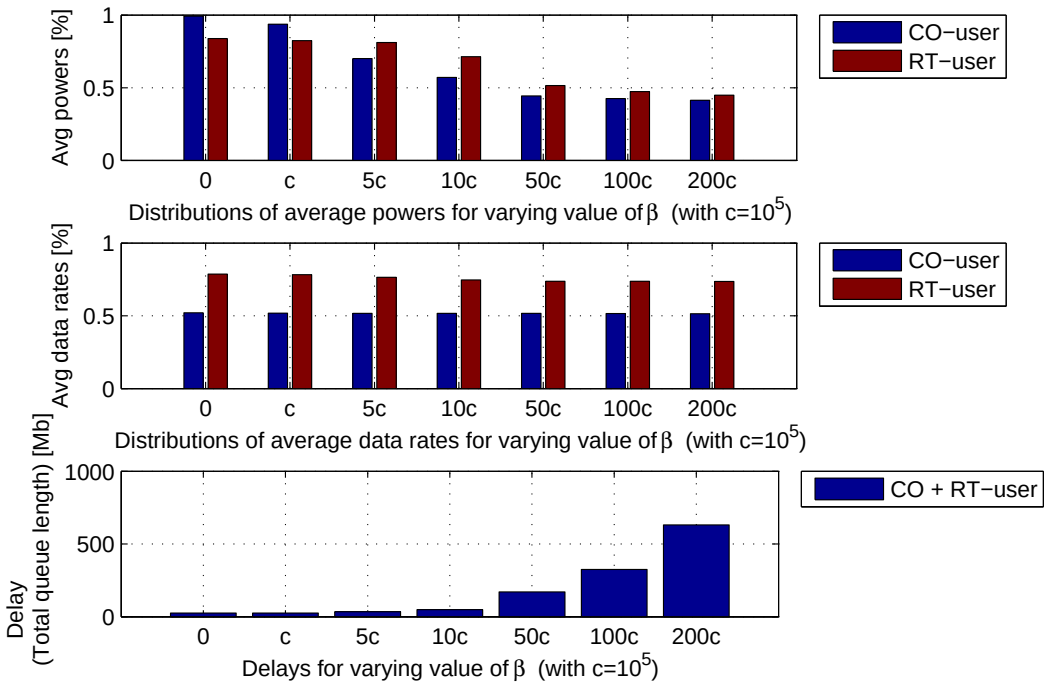


Fig. 5. Influence of parameter β of (13) on average powers, average data rates and delay performance for 2-user ADSL scenario.

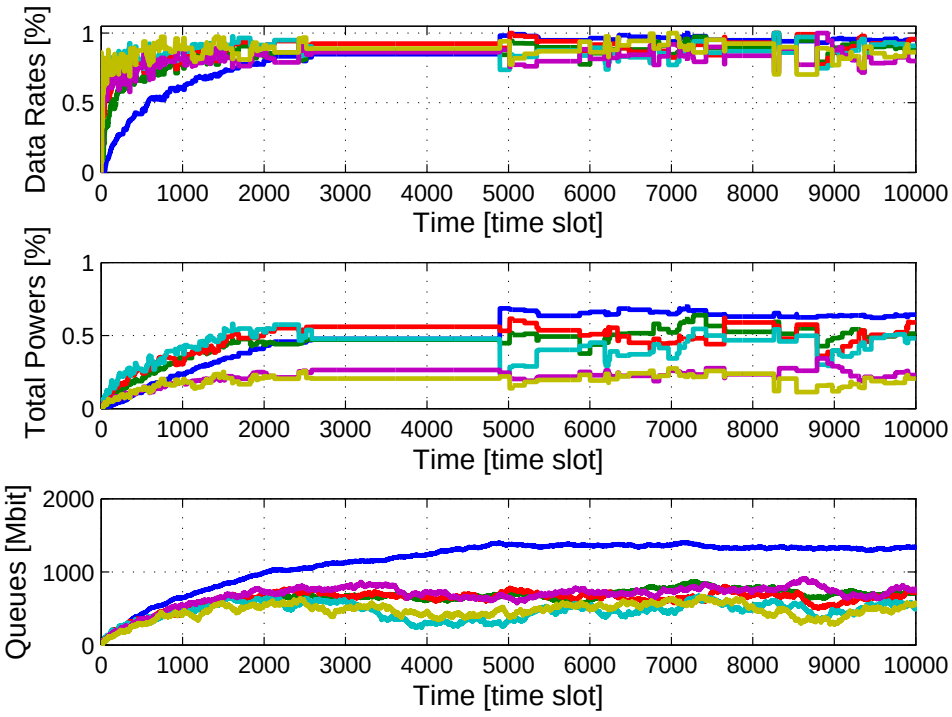


Fig. 6. Convergence behaviour and variation of the data rates, total transmit powers and queue lengths for 6-user ADSL scenario of Figure 1. Modems 1 to 6 correspond to the colors blue, green, red, cyan, magenta and yellow, respectively.

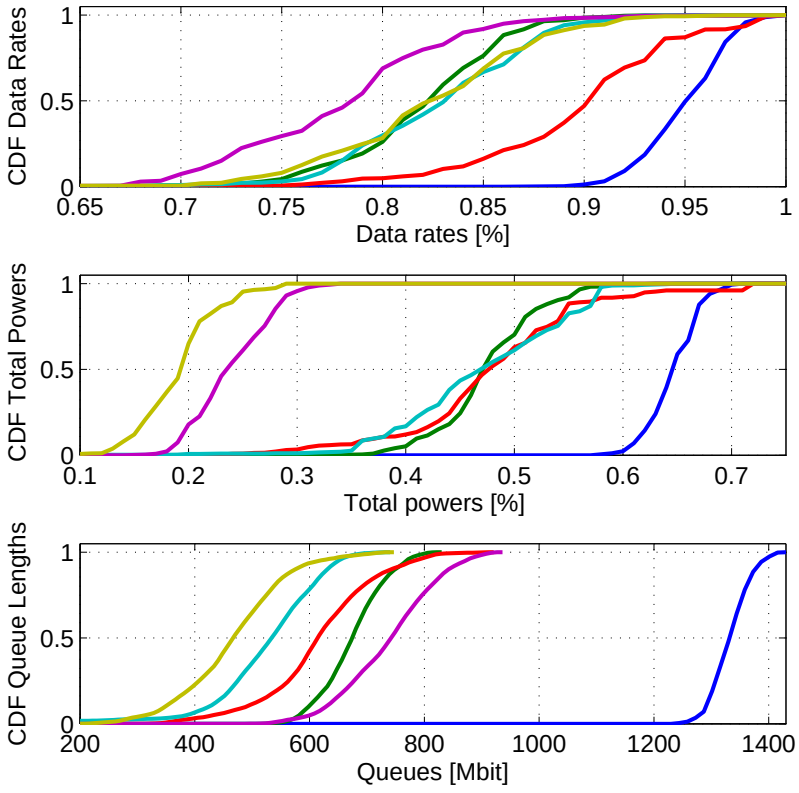


Fig. 7. Cumulative probability distribution (CDF) of data rates, total powers, and queue lengths in steady state operation for 6-user ADSL scenario of Figure 1. Modems 1 to 6 correspond to the colors blue, green, red, cyan, magenta and yellow, respectively