Throughput and Delay of DSL Dynamic Spectrum Management with Dynamic Arrivals

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

***Abstract*—In modern DSL networks, crosstalk among different lines (i.e., users) is the major source of performance degradation. Dynamic Spectrum Management (DSM) refers to a set of techniques to mitigate the effect of crosstalk leading to spectacular performance gains. However the main research efforts in DSM aim at only physical layer performance whereas the true end user experience depends on what they see at the application rather than the physical layer. Upper layer performance metrics like throughput and delay may be much more important to improve the user satisfaction. To that end, we provide a framework to study upper layer performance by looking at scheduling and DSM together. We show how optimal scheduling can be combined with optimal DSM and provide throughput-optimal scheduling algorithms which require only polynomial complexity. We furthermore present extensions that significantly improve delay performance by using the specific structure of the underlying problem.**

I. Introduction

Digital Subscriber Line (DSL) technology provides broadband access over twisted pair copper wires of the existing telephone network. Nowadays it is still the most popular broadband access technology worldwide. The major obstacle for performance improvement remains the excessive electromagnetic interference, also called crosstalk, generated among different lines in the same cable bundle.

One promising set of techniques for tackling the crosstalk problem is Dynamic Spectrum Management (DSM). The main idea of DSM is to prevent and/or remove crosstalk by spectrum and/or signal coordination, respectively. In this paper, we focus on spectrum level coordination, also referred to as spectrum management, spectrum balancing, or multi-carrier power control. From an information-theoretic point of view, this is known as a multi-carrier interference channel where the interference from the other users is treated as noise. Many DSM algorithms[1] are proposed in literature ranging from fully autonomous [1], [2] and distributed [3]–[5] to centralized algorithms [6]–[8].

The major research efforts in DSM algorithms aim at only physical layer performance, which is to maximize the data rates subject to power constraints. However the end user experience depends on what they see at the application rather than physical layer. The application workload naturally arrives with only a finite workload instead of the infinite workload assumed by physical layer DSM design. Furthermore an increasing amount of applications (e.g. video and voice) are delay-sensitive. Motivated by the above two that have not been treated in existing and vast DSM literature, we look at how upper layer scheduling and physical layer DSM can work together so as to incorporate upper layer performance metrics such as throughput and delay. As we show later, both optimal scheduling and DSM are computationally intractable, i.e., NP-hard. This paper tries to address the questions of how relaxations on both problems work well together towards good upper layer performance.

The main contributions of this paper are as follows:

(i) We provide a generalized framework to study the upper layer performance over DSL networks, motivated by the recent research in other areas, e.g., wireless scheduling. This framework is a useful tool to understand throughput and delay performance provided by DSM algorithms at the physical layer, as well as gives us practical implications on the design of future DSM algorithms. The proposed framework facilitates characterization of throughput and delay properties, and also discloses the generic trade-off between complexity, and throughput or delay.

(ii) Using this framework, we connect throughput-optimal scheduling to the globally optimal DSM algorithms that require exponential computational complexity. However, somewhat surprisingly, we show that it is possible to achieve optimal throughput by using (randomized) sub-optimal DSM algorithms with only polynomial time complexity. However, the price to pay is increasing delay.

(iii) Then, we provide algorithms which significantly improve delay performance with only small extra complexity, by exploiting the specific structure of our problem, namely correlation in frequency and time domain, and also single user optimality for large crosstalk DSL scenarios.

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-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) to the user $n, \in N$, at time t. The arrival process is assumed to be i.i.d. across time-slots, where $\mathbb{E}[A^n(t)] = \lambda^n$. We assume that the duration of one time-slot is small enough so that λ^n is upper bounded by a constant, denoted by Ω, i.e., $A^n(t) \leq \Omega$, $\forall n \in N$, $\forall t \geq 0$. Denote by $\boldsymbol{\lambda} = (\lambda^n : \; n \in N)$ the (mean) arrival vector across users.

Resource Model. The network resources are represented by a finite set $\mathcal{R}$ of the feasible rate vectors, referred to as

P. Tsiaflakis and M. Moonen are with Electrical Engineering, Katholieke Universiteit Leuven, Belgium, and Y. Yi and M. Chiang with are Electrical Engineering, Princeton University, USA. Paschalis Tsiaflakis is a research assistant with the F.W.O. Vlaanderen.

[1]In this text DSM refers to spectrum balancing.

rate region describing the simultaneous 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. This results in the following characterization of the achievable rate region $\mathcal{R}$:

$$\mathcal{R}=\Big\{(R^n:n\in N)|R^n=f_s\sum_k b_k^n,\sum_{k\in K}s_k^n\le P^n,0\le s_k^n\le s_k^{n,\mathrm{mask}}\Big\},$$

where s_k^n denotes the transmit power of user n on tone k, P^n denotes the total power budget available to user n, $s_k^{n,\mathrm{mask}}$ denotes the spectral mask constraint for user n on tone k, and f_s denotes the DMT symbol rate. The b_k^n, which depends on the transmit powers, denotes the bit rate of user n on tone k, given by:

$$b_k^n \triangleq \log_2\left(1+\frac{1}{\Gamma}\frac{|h_k^{n,n}|^2 s_k^n}{\sum_{m\neq n}|h_k^{n,m}|^2 s_k^m+\sigma_k^n}\right)\ \text{bits/s/Hz},\quad (1)$$

where $[\mathbf{H}_k]_{n,m}=h_k^{n,m}$ is an $N\times N$ matrix containing the channel transfer functions from transmitter m to receiver n on tone k. The diagonal elements are the direct channels, the off-diagonal elements are the crosstalk channels. σ_k^n denotes the noise power that contains thermal noise, alien crosstalk and radio frequency interference (RFI). Note that the channel and noise are characterized as constant over time but it can simply be extended to be varying in function of time slots. Γ denotes 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 [9].

Resource Allocation Scheme. Then, a resource allocation scheme chooses a sequence of rate schedules ($\boldsymbol{R}(t) = (R^n(t): n\in N))_{t=0}^{\infty}$, $\boldsymbol{R}(t)\in\mathcal{R}$. Since the main resources of our system are the transmit power spectra $(s_k^n(t): n\in N, k\in K)$ and the rate schedule $\boldsymbol{R}(t)$ is determined by the allocated power spectra, we use "resource allocation" algorithm and "(power) scheduling" algorithm interchangeably throughout the paper.

Queueing Dynamics. Denote by $Q^n(t)$ the queue length of the buffer of user n at time t. The queueing dynamics are then defined by the following recursion:

$$Q^n(t+1) = \big[Q^n(t)-R^n(t)\big]^+ + A^n(t+1),\quad (2)$$

where $[x]^+=\max(x,0)$.

B. Performance Metrics

Performance metrics considered in this paper are throughput and delay that may be the two performance goals in networking and communication systems. To define throughput, we first define the notion of stability that essentially represents the condition that queue-lengths remain finite.

Definition II.1 (Stability)**.** *The system is said to be stable, if*

$$\lim\sup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t}\mathbb{E}\left[\sum_{n\in N}Q^n(\tau)\right]<\infty.$$

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

Definition II.2 (Throughput-region)**.** *The throughput-region $\Lambda\subset\mathbb{R}_+^N$ is the set of all arrival vectors for which there exists a scheduling algorithm stabilizing the system.*

Throughput-region can be regarded as the maximum throughput than one can achieve. In other words, no scheduling can stabilize the systems for the arrival rate vector outside of the throughput-region. The throughput-region can be characterized as: Λ = convex-hull($\mathcal{R}$). We say that a scheduling which can stabilize the system for any arrival rate vector in Λ is *throughput-optimal.*

In addition to throughput, we also consider the total (stationary) queue length over users, given by: $\sum_{n\in N}\mathbb{E}[Q^n(\tau)]$. This naturally relates to delay from Little's law.

Our objective is to develop a resource allocation scheme for this coupled N-user DSL system, that achieves throughput-optimality *without* knowledge of the mean arrival rates.

III. Throughput-Optimal Scheduling and DSM Algorithms

Conventional DSM algorithms aim at optimizing physical layer performance, i.e. allocate transmit powers so to maximize data rates subject to power constraints. Here we will extend the objective to upper layer performance metrics such as throughput and delay.

We first describe a scheduling that is throughput-optimal, and then show its connection to the conventional DSM problem formulation. To that end, we first define the *weight* of the rate schedule $\boldsymbol{R}(t)$ as: $W(t)\triangleq W(\boldsymbol{R}(t))\triangleq\sum_{n\in N}Q^n(t)R^n(t)$. Consider the following scheduling, referred to as Max-Weight (MW): at time-slot t, it schedules $\boldsymbol{R}^\star(t)$ that maximizes the weight, i.e.,

$$\boldsymbol{R}^\star(t)=\arg\max_{\boldsymbol{R}\in\mathcal{R}}\sum_{n\in N}Q^n(t)R^n.$$

It has been proved that MW scheduling is throughput-optimal under slightly different system models (e.g., [10]), and extension to our system model is straightforward. By incorporating the DSM physical layer resources introduced in section II-A, it is easy to observe that MW scheduling comes down to solving the following highly nonconvex optimization problem at every time-slot t:

$$\max_{s_1^1,\dots,s_K^N}\sum_{n=1}^{N}Q^n(t)R^n(t)$$
$$\text{s.t.}\ \sum_{k=1}^{K}s_k^n(t)\le P^n,\ 0\le s_k^n(t)\le s_k^{n,\mathrm{mask}},\ \forall n,k.\ (3)$$

Note that the objective of the existing (rate-adaptive) DSM algorithms in literature is to solve the optimization problem in (3), where the queue length of the user n is used as its weight[2], i.e., $w^n=Q^n(t)$ (see e.g., [6]). In other words, existing DSM algorithms can be reused as an underlying building block of Max-Weight scheduling, where the DSM problem has to be solved at each time slot, replacing w^n by the queue lengths $Q^n(t)$ at time slot t.

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

At this point we distinguish three different types of DSM algorithms that are (i) globally optimal, (ii) locally optimal, and (iii) heuristic. Globally optimal DSM algorithms, such as OSB [6], BB-OSB [7] and PBB [8], succeed in finding the globally optimal solution of (3) where the feasible set of transmit power spectra is discretized up to accuracy Δ_s. Locally optimal DSM algorithms, such as SCALE [3], DSB [4] and MIW [5], only guarantee a locally optimal solution to (3). Heuristic DSM algorithms, such as IW [1] and ASB [2], do not necessarily ensure globally or locally optimal solutions to (3), but have practically strong merits in near-optimality with reasonably low complexity.

Thus, scheduling based on globally optimal DSM algorithms can be regarded as throughput-optimal algorithms, i.e., achieving maximum throughput performance. However, these algorithms require computationally intractable complexity (i.e., NP-hard in terms of N), leading to difficulty in practical implementation. Our question is whether we can achieve throughput-optimality even with sub-optimal DSM algorithms having lower, practical complexity. In this paper, due to space limitations, we restrict our focus on just locally optimal algorithms, and further we do not differentiate centralized and distributed algorithms. To formally study, we first introduce randomized scheduling algorithms in the next section.

IV. Polynomial Complexity Scheduling Achieving Throughput Optimality

Locally optimal DSM algorithms [3]–[5] require only polynomial complexity. However they sometimes fail to find the globally optimal solution of the nonconvex problem (3). Indeed, their final solution strongly depends on the chosen *initial starting point*. Some initial starting points lead to the globally optimal solution whereas others lead to only a locally optimal solution. We note that there always exist at least one initial starting point or a region of starting points leading to globally optimality. We will exploit this starting point dependence in combination with an appropriate scheduling scheme to design polynomial complexity throughput-optimal scheduling algorithms.

A. δ-Randomized DSM Algorithms

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

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

$$\mathbb{P}[\boldsymbol{R}'(t) == \boldsymbol{R}^\star(t) \mid \boldsymbol{Q}(t)] \geq \delta, \tag{4}$$

where $\boldsymbol{R}^\star(t)$ is the optimal solution of (3).

In other words, a δ-randomized DSM algorithm randomly generates a rate schedule $\boldsymbol{R}'(t)$ that is guaranteed to be equal to the optimal rate schedule with positive probability. Clearly, globally optimal DSM algorithms are 1-randomized DSM.

Now, consider the following scheduling algorithm using a δ-randomized DSM, referred to as *Pick-and-Compare*. Pick-and-Compare scheduling can be interpreted as a randomized scheduling that produces a "reasonably good" schedule in terms of its non-zero probability of finding a globally optimal schedule (**Step 1**), in conjunction with progressively selecting better schedules by comparing the previous schedule and the current randomized schedule (**Step 3**). Again, when $\delta = 1$, the Pick-and-Compare scheduling recovers the Max-Weight scheduling algorithm. Theorem IV.1 states the throughput property of the Pick-and-Compare scheduling.

Algorithm 1 Pick-and-Compare Scheduling: at time-slot t

Step 1 "Pick" a random rate schedule $\boldsymbol{R}'(t)$ using δ-randomized DSM.
Step 2 Compute the weight of $\boldsymbol{R}'(t)$, i.e., $W(\boldsymbol{R}'(t))$.
Step 3 "Compare" $W(\boldsymbol{R}'(t))$ and $W(\boldsymbol{R}(t-1))$, and take the rate schedule with larger weight for the rate schedule at slot t, i.e., $\boldsymbol{R}(t) = \arg\max_{\boldsymbol{S}\in\{\boldsymbol{R}'(t),\boldsymbol{R}(t-1)\}} W(\boldsymbol{S})$.
Step 4 Apply $\boldsymbol{R}(t)$ to transmit data.

Theorem IV.1. *For any $0 < \delta \leq 1$, a Pick-and-Compare scheduling with δ-randomized DSM is throughput-optimal.*

The same result has been proved under different system models (see e.g., [11], [12] in wireless scheduling). Thus, we omit the complete proof due to space limitations, but provide only sketch of the proof: The stochastic stability of the considered queueing system can be proved by defining a certain Lyapunov function and then proving that the system has a negative Lyapunov drift whenever the aggregate queue-length is greater than some constant B. The throughput-optimality of MW scheduling can be proved by defining a quadratic Lyapunov function, $L(t) \triangleq \sum_{n\in N} Q^n(t)^2$. In fact, we can easily show that Pick-and-Compare scheduling satisfies the following: $W(t) \geq W^\star(t) - C(t)$, where $E[C(t)] \leq K$, for some constant K. The (random) suboptimality $C(t)$ is due to the fact that Pick-and-Compare can find the maximum-weight scheduling only every $1/\delta$ slot on average, and this additive suboptimality does not affect the stability region.

These δ-randomized algorithms indeed represent 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 leads to algorithms with lower complexity at the cost of increasing delays, as discussed in Sections IV-B and IV-C.

Now, we provide a simple way of realizing δ-randomized DSM using the existing locally optimal DSM algorithms. This is based on random selection of an initial starting point. We first denote by $\mathcal{T}$ the set of all possible transmit power combinations discretized up to accuracy Δ_s:

$$\mathcal{T} = \{(y_{ij} : i \in N, j \in K) | y_{ij} \in \{0, \Delta_s, \ldots, s_j^{i,\max}\}\}, \tag{5}$$

where $s_j^{i,\max} \triangleq \min(P^i, s_j^{i,\text{mask}})$. Note that the globally optimal solution $\boldsymbol{s}^\star$, obtained by the globally optimal DSM algorithms, belongs to this set. By randomly choosing an element of $\mathcal{T}$ and using it as the initial transmit power vector for the locally optimal DSM algorithm, there should be a non-zero probability to converge to globally optimal performance. More specifically, if the optimal solution is chosen from $\mathcal{T}$ (i.e., we allow only discretized powers in the system), it converges to the global-optimal solution as the locally optimal DSM algorithms are monotonically increasing over their successive iterations. This leads to a δ-randomized DSM algorithm, which will be referred to as **R1-DSM**. Then, **R1-DSM** can be used in combination with the Pick-and-Compare scheduling of Algorithm 1 to obtain a throughput optimal DSM scheduling scheme with only polynomial complexity, where the lower-bound on δ is $\frac{1}{|\mathcal{T}|}$.

R1-DSM (Random Initial Point)

1) Pick a random initial transmit power $\boldsymbol{x} \in \mathcal{T}$ uniformly,
2) Apply a locally optimal DSM algorithm starting with $\boldsymbol{x}$.

B. Delay Performance of δ-Randomized DSM

As shown in the previous section, it is possible to achieve throughput optimality with randomized locally optimal DSM algorithms. What price should be paid for this complexity reduction from exponential to polynomial without losing throughput region? Next, in Section IV-C we will show that the price is *delay*. Further, we use this characterization of price to develop other scheduling algorithms with polynomial complexity that improve delay performance with small extra complexity.

Calculating the exact delay performance in our system is known to be very difficult, still an open problem mainly due to 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, this bound is quite helpful to understanding how delay performance scales with δ, which in turn relates to the complexity of scheduling algorithms.

The delay performance should depend on the input arrival rate $\boldsymbol{\lambda}$, e.g., as $\boldsymbol{\lambda}$ 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 vector and the throughput region as follows:

Definition IV.2 (Distance)**.**

$$d(\boldsymbol{\lambda}) = \sup\{\delta : \boldsymbol{\lambda} \in (1-\delta)\Lambda\}. \tag{6}$$

This distance essentially represents how heavily the system is loaded, where smaller $\boldsymbol{\lambda}$ leads to larger $d(\boldsymbol{\lambda})$. Using this definition, the following bound on the delay can be obtained for randomized scheduling with δ-randomized DSM algorithm and $\boldsymbol{\lambda} \in \Lambda$.

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{n\in N}\mathbb{E}\Big[Q^n(t)\Big] \quad \leq \quad \frac{N^2\Omega(2/\delta+1)}{d(\boldsymbol{\lambda})},$$

where recall that Ω is the maximum number of arrivals at one slot. We can observe that as δ decreases, the delay bound increases. We skip the proof for space limitation. The idea is to define a quadratic Lyapunov function, and to telescope its drift across time-slots (see [13] for details).

Now, let us investigate the delay performance of Pick-and-Compare with **R1-DSM**. As discussed earlier, $\delta \geq \frac{1}{|\mathcal{T}|}$, where the delay bound becomes very large, i.e., delay bound $\approx O(|\mathcal{T}|)$. This indicates that the delay of **R1-DSM** is pretty large in the worse-case sense. However, note that in many cases, δ may be much larger since depending on the scenario, we may have many possible initial starting points converging to the global optimal solution. Next, we will propose a modified DSM that improves the delay performance significantly with small extra complexity.

C. Significant Delay Decrease with Small Extra Complexity

Locally optimal DSM algorithms reduce complexity, which, instead, guarantees finding the global optimal solution only probabilistically (in conjunction with a random initial point) with negative effect on delay. Can we improve delay performance significantly (i.e., increase the probability of finding the globally optimal solution), by exploiting the specific structure of our problem? The answer is positive, and we will use the following specific features to achieve the goal:

1) **Time-slot correlation**: When queue lengths are large (i.e., the input arrival vector is close to the boundary of throughput-region), the system does not observe much difference in weights (i.e., queue lengths) over subsequent time-slots.
2) **Tone correlation**: Subsequent tones have similar channel characteristics and so also have similar solutions.

These problem-specific correlations across time-slot and tones can actually be exploited to choose the initial starting points more intelligently. One can add two additional initial starting point to the random point in each tone: (1) the best local optimum at that tone from the previous time slot using the idea of *time-slot correlation*, and (2) the best local optimum from the previous tone using the idea of *tone correlation*. The inclusion of these extra initial points will increase δ, resulting in better delay performance. This leads to the following δ-randomized DSM algorithm that can be combined with the Pick-and-Compare in Algorithm 1 to obtain a throughput-optimal scheduling with significantly improved delay performance, compared to **R1-DSM**.

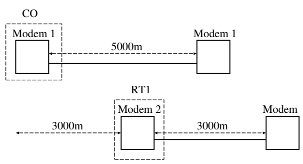
R2-DSM (Time-slot and tone correlation)

1) For each tone, choose three initial starting points: *i)* random point, *ii)* the best local optimum at that tone from the previous time slot, *iii)* the best local optimum from the previous tone,
2) Apply a locally optimal algorithm for all initial starting points and retain best converged local optimum in each tone.

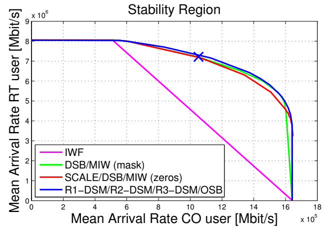
A third δ-randomized DSM is inspired by the recent result in [14], where it is stated that for large crosstalk scenarios the solution of (3) is an FDMA solution, i.e. a solution where only one user is active at each tone. We indeed have observed that for large crosstalk scenarios the locally optimal solutions get isolated along the axes [4]. 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 $s_k^{n,\mathrm{mask}}$. By adding these initial points, the likelihood that one of these initial starting points leads to the globally optimal solution of that tone, increases significantly. This results in the following δ-randomized DSM algorithm:

R3-DSM (N single-user initial points)

1) For each tone, choose $N+1$ initial starting points: *i)* random point, *ii)* N single user initial starting points at spectral mask,
2) Apply locally optimal algorithm for all initial starting points and retain best converged local optimum in each tone.

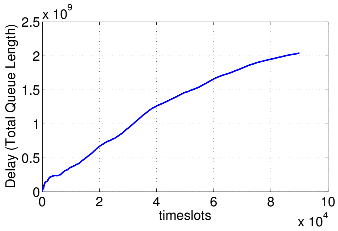


(a)

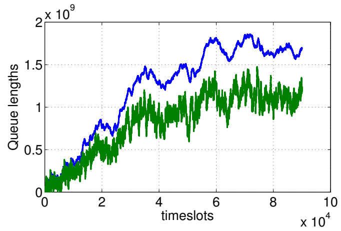


(b)

Fig. 1. Simulation Scenario and Stability Region. The curve indicated by DSB/MIW (mask) refers to the stability region of the DSB and MIW algorithms with fixed initial starting point equal to spectral mask, i.e. $s_k^n = s_k^{n,\mathrm{mask}}, \forall n, k$. The curve indicated with SCALE/DSB/MIW (zeros) starts from the all zero starting point.

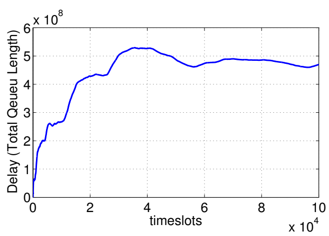


(a) Aggregate total queue length

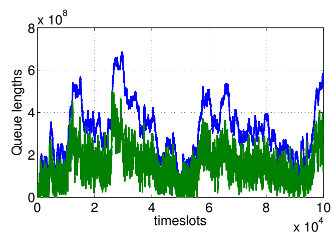


(b) Instantaneous queue lengths [bits]

Fig. 2. Stability and delay behaviour of Pick-and-Compare with DSB (zeros) algorithm;for (b): (blue: CO user, green: RT user)



(a) Aggregate total queue length



(b) Instantaneous queue lengths [bits]

Fig. 3. Stability and delay behaviour of Pick-and-Compare with **R2-DSM**; for (b): (blue: CO user, green: RT user)

Note that all of **R1-DSM**, **R2-DSM**, and **R3-DSM** provably achieve the maximum throughput. We observe that there is a clear trade-off between complexity and delay. 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, inverse proportional to the delay, we have that $\delta_{\mathrm{R1-DSM}} \leq \delta_{\mathrm{R2-DSM}} \simeq \delta_{\mathrm{R3-DSM}}$. In fact for very nonconvex DSL scenarios 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 **R2-DSM** and **R3-DSM** for just a slight increase in complexity.

V. Simulation Results

This section presents simulation results of throughput and delay properties of the proposed algorithms for a realistic ADSL downstream scenario shown in Figure 1(a). The twisted pair lines have a diameter of 0.5 mm (24 AWG). The maximum transmit power is 20.4 dBm. The SNR gap Γ is 12.9 dB. The tone spacing Δ_f is 4.3125 kHz. The DMT symbol rate f_s is 4 kHz. Note that for most of the DSL scenarios, locally optimal algorithms perform globally optimal. For this reason we intentionally chose to focus on the CO-RT scenario of Figure 1(a), which is known as a very difficult scenario where locally optimal algorithms generally fail to find the globally optimal solution.

In Figure 1(b) we show the stability regions for various algorithms including randomized algorithms proposed in this paper. Note that the throughput of the randomized DSM algorithms is optimal in contrary to the algorithms with fixed initial points, whose stability regions are sub-optimal and depend on the initial starting points. In Figures 2 and 3 we show the delay performance and instantaneous queue lengts of Pick-and-Compare scheduling combined with DSB (or similarly for MIW) with fixed zeros initial points and **R2-DSM**, respectively, for the mean arrival rate indicated with the cross in Figure 1(b). One can see that **R2-DSM** succeeds in stabilizing the system with only small delay, whereas the queues for DSB keep increasing after 100000 time slots. Finally we would like to remark that the delay performance of **R2-DSM** and **R3-DSM** is much better with respect to **R1-DSM**, as we analyzed in the delay bound.

VI. Conclusion

It is crucial to understand and design algorithms for DSL systems from the perspective of QoS that will be experienced by the users. In this paper, we show that the results from physical-layer DSM, under assumption of infinite backlog, may be significantly changed. More specifically, by jointly considering upper layer scheduling and the physical-layer DSM algorithm, somewhat surprisingly, even sub-optimal DSM algorithms can achieve throughput maximization, when intelligent scheduling algorithms are employed, with an interesting trade-off between complexity and delay.

References

[1] W. Yu, G. Ginis and J. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE J. Sel. Area. Comm.*, vol. 20, no. 5, pp. 1105–1115, Jun. 2002.

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

[3] J. Papandriopoulos and J. S. Evans, "Low-complexity distributed algorithms for spectrum balancing in multi-user DSL networks," in *Proceedings of IEEE ICC*, 2006.

[4] P. Tsiaflakis and M. Moonen, "Low-complexity dynamic spectrum management algorithms for digital subscriber lines," in *Proceedings of ICASSP*, 2008.

[5] W. Yu, "Multiuser water-filling in the presence of crosstalk," in *In Proceedings of Information Theory and Applications (ITA)*, 2007.

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

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

[8] Y. Xu, S. Panigrahi, and T. Le-Ngoc, "A concave minimization approach to dynamic spectrum management for digital subscriber lines," in *Proceedings of IEEE ICC*, 2006.

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

[10] 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.

[11] 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.

[12] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proceedings of ACM Sigmetrics*, 2007.

[13] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[14] S. Hayashi and Z. Q. Luo, "Dynamic spectrum management: When is FDMA sum-rate optimal?" in *Proceedings of ICASSP*, 2007.