♦ Performance of Digital Subscriber Line Spectrum Optimization Algorithms

Miroslav Živković, Gerhard Kramer, Carl J. Nuzman, Carl Posthuma, James Wheeler, Phil Whiting, and Adriaan J. de Lind van Wijngaarden

A management standard for digital subscriber line (DSL) systems has been defined in order to ensure the spectral compatibility of the signals, services, and technologies that are deployed. Compatibility is typically accomplished by using static spectral masks. This conservative approach may lead to suboptimal performance of DSL systems. Recently, a number of dynamic spectrum management (DSM) solutions were proposed that improve DSL performance by adaptive application of spectral masks, also known as DSM level 2. The masks are calculated taking into account actual performance requirements and impairments, such as crosstalk. We present improved DSM level 2 algorithms that are faster than existing algorithms and demonstrate potential rate/reach gains by numerical simulations for some DSL technologies, such as asymmetric digital subscriber line transceivers. We also present a solution for flexible deployment of DSM spectrum optimization algorithms in the field. © 2008 Alcatel-Lucent.

Introduction

The rates of deployed digital subscriber line (DSL) systems are constrained by a variety of impairments. These include noise impairments such as background noise, impulse noise, and radio frequency interference (RFI), as well as interference impairments due to crosstalk between lines. Crosstalk may be orders of magnitude larger than any other type of noise in a DSL system.

One particular measurement of far-end crosstalk (FEXT) as a function of signal frequency is shown in **Figure 1**. The measurement was performed for the frequency range up to 3.5 MHz on an American wire gauge-24 (AWG-24) cable and for a binder of six AWG-24 cables in total. These and other measurements indicate that crosstalk is a complex phenomenon that depends, among other factors, upon

- Frequency,
- Proximity of disturber and victim line within the binder,
- Distance from the disturber's transmitter to the point at which crosstalk is observed,
- Coupling length,
- Temperature, and
- Humidity.

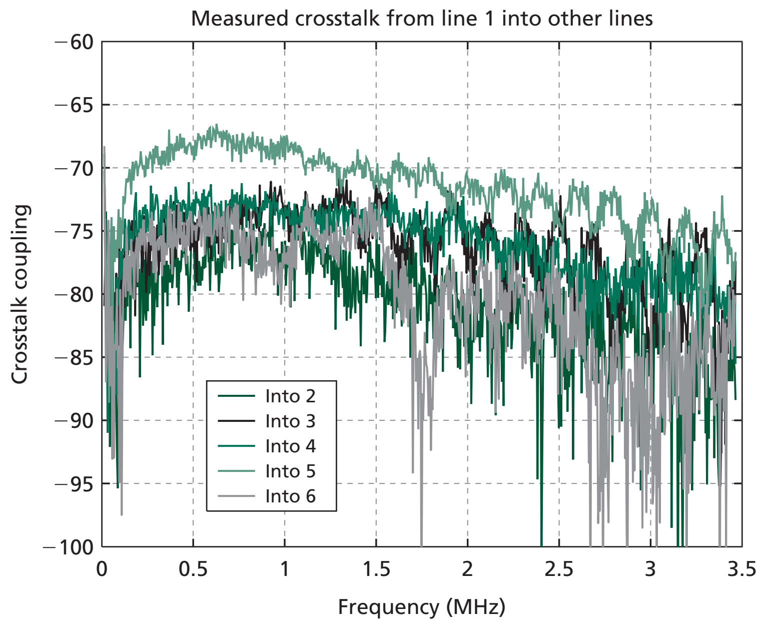
The current American National Standards Institute (ANSI) T1.417 Spectrum Management Standard [2] contains a model which estimates the "1 percent worst case" crosstalk between any two pairs in the binder.

Bell Labs Technical Journal 13(1), 129–146 (2008) © 2008 Alcatel-Lucent. Published by Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com) • DOI: 10.1002/bltj.20287

Panel 1. Abbreviations, Acronyms, and Terms

ADSL—Asymmetric digital subscriber line
ADSL2+—Asymmetric digital subscriber line transceivers—extended bandwidth ADSL2
ANSI—American National Standards Institute
ASB—Autonomous spectrum balancing algorithm
AWG—American wire gauge
CO—Central office
CPE—Customer premises equipment
DELT—Dual ended line testing
DMT—Discrete multi-tone
DSL—Digital subscriber line
DSLAM—Digital subscriber line access multiplexer
DSM—Dynamic spectrum management
FEXT—Far-end crosstalk
FTP—File Transfer Protocol
GUI—Graphical user interface
IEEE—Institute of Electrical and Electronics Engineers
IP—Internet Protocol
ISB—Iterative spectrum balancing algorithm
IWF—Iterative water-filling algorithm
MIB—Management information base
MVC—Model-view-control
OSB—Optimal spectrum balancing algorithm
PC—Personal computer
PSD—Power spectrum density
RFI—Radio frequency interference
RT—Remote terminal
SCALE—Successive convex approximation for low-complexity algorithm
SMC—Spectrum management center
SNMP—Simple Network Management Protocol
SNR—Signal-to-noise ratio
UM—Utility maximization algorithm

The model takes as input quantities that are known in principle, such as line lengths, while positioning unknown quantities in the binder to assume worst case values. The worst case, 1 percent crosstalk, is based on elaborate measurement campaigns [11, 13] which led to the development of a model that ensures that the level of crosstalk for 99 percent of deployed DSL lines is smaller than the one specified by the



FEXT—Far-end crosstalk

Figure 1.
One particular measurement of FEXT as a function of signal frequency.

model. This empirical model was adopted by the standardization bodies because the crosstalk properties between any line pairs are perceived to be hard to measure and to act upon. As a consequence of this decision, the current spectrum management practices are conservative and static. They are typically used to apply a set of static rules to the power spectral density (PSD) masks in actual deployed systems. Further, any new xDSL technology has to demonstrate spectrum compatibility with all previously deployed xDSL technologies.

The current (static) spectrum management rules ignore the fact that the behavior of DSL loops in any deployed binder is dynamic. In other words, the modems are (de)activated or re-trained at random times. Dynamic spectrum management (DSM) techniques include a number of solutions that aim to dynamically optimize the DSL communication with respect to the channel and transmission settings of the neighboring modems within the same binder. DSM is tailored to the actual crosstalk environment and crosstalk couplings between the individual lines in a particular cable. Depending on the level of coordination among the modems that are using the lines within the same binder, the solutions are usually categorized as DSM levels 0, 1, 2, and 3, with level 3 requiring the greatest coordination [1]. Another popular classification methodology categorizes DSM techniques as either spectrum coordination techniques (which include DSM level 0, 1, and 2 solutions) or signal coordination techniques (DSM level 3 solutions). Further classifications may distinguish between distributed versus centralized coordination algorithms.

The first generation of DSM solutions promised large improvements in bit rates and range. The improvements were not surprising, since, e.g., single-disturber crosstalk is typically 10 dB to 20 dB smaller than the corresponding 1 percent worst case model. However, the gains were demonstrated for scenarios involving at most four lines [3, 4, 8]. The reason for selecting such a limited scenario is that some of the applied spectrum optimization algorithms are not computationally efficient: i.e., their complexity grows exponentially with the number of lines or tones analyzed. Furthermore, typical 25-line or 50-line binder crosstalk is 3 dB to 6 dB smaller than the corresponding 1 percent worst case model [6, 9]. This implies that improvements due to spectrum optimization are likely to be less spectacular than predicted. DSM algorithms with improved performance and reduced complexity are required to calculate the possible improvements for binders such as these.

In this paper, we present computationally efficient DSM spectrum optimization algorithms and analyze their performance. In addition, we propose a system architecture that supports fast deployment of DSM spectrum optimization in the field.

Overview of Spectrum Management Algorithms

Spectrum optimization algorithms dynamically allocate transmit spectra for discrete multi-tone (DMT) DSL systems. The allocation is based upon the level of crosstalk, the deployment topology (e.g., central office [CO] or remote terminal [RT] deployment), and the level of background noise.

The objective of spectrum management algorithms is to find optimal PSD masks for N modems, $P_1, P_2, \ldots, P_N$ (where P_i is the PSD mask for user i) in order to maximize the rates of the modems $R_1, R_2, \ldots, R_N$. The rate for user n is given by

$$R_n = \sum_{k=1}^{K} R_n^k \; (P_n^k) = f_s \sum_{k=1}^{K} \log\left(1 + \frac{|h_{n,n}^k|^2 P_n^k}{\Gamma\left(\sum_{l \neq n} |h_{n,l}^k|^2 P_l^k + \sigma_k^2\right)}\right) \tag{1}$$

where f_s is the symbol rate, $h_{n,n}^k$ is the direct channel coefficient for user n at tone k, P_n^k is the power used by user n at tone k, Γ is the coding gain, $h_{n,l}^k$ is the crosstalk coefficient from disturber l into victim line n on tone k, and σ_k^2 is the noise power on tone k. This maximization is usually subject to constraints on the total power assigned to each user and on spectral masks.

A number of DSM algorithms have been developed. One of the first algorithms was iterative waterfilling (IWF) [14]. Waterfilling assigns power to parallel Gaussian channels in order to maximize the sum of the channel rates, subject to a total power constraint. IWF is an autonomous, low complexity, distributed greedy optimization algorithm in which each user in turn uses waterfilling to compute the power spectrum that maximizes its own rate, taking into account interference from all other active modems in the binder. IWF is in general suboptimal except for the case of a single user, because in greedily maximizing its own rate, a user does not take into account the degree to which the rates of other users may be reduced.

The optimal spectrum balancing (OSB) [5] and iterative spectrum balancing (ISB) algorithms [4] try to maximize a weighted sum $\sum_n w_n R_n$ of user rates subject to the power constraints. Lagrange multipliers representing the power constraints are used to decouple the optimization problems across tones. In OSB, the optimal power allocation for each tone is determined by exhaustive search on a discrete grid. Although it is optimal, the OSB algorithm suffers from exponential complexity with the number of users, which makes it feasible only for a few users. The ISB algorithm is a near-optimal algorithm with reduced complexity. As with IWF, each iteration of the algorithm modifies the spectrum of one user while holding all other users' spectra fixed. However, instead of greedily maximizing the rate of one user, ISB maximizes the weighted sum of all user rates. Because the complexity of ISB is polynomial in the number of users, it is still difficult to implement in real time for a large number of users.

The successive convex approximation for low-complexity (SCALE) algorithm [12] uses the same weighted rate sum rate objective as OSB and ISB. Like ISB, this algorithm always converges to a local optimum and is reported to be nearly optimal in most cases. The algorithm, based on geometric programming, has relatively low complexity and can in principle be implemented in a distributed manner via message passing between the customer premises equipment (CPE) and the centralized spectrum management center (SMC) connected to the digital subscriber line access multiplexer (DSLAM).

One of the most recently developed algorithms is the autonomous spectrum balancing (ASB) algorithm [8]. In this approach, a fictitious "reference" user is defined, and each user autonomously attempts to balance the goals of increasing its own rate and increasing the rate of the reference user. This algorithm has low complexity and is well suited to distributed implementation but does not guarantee optimal results.

In all of the problems based on maximizing the weighted sum of rates, the overall problem to be solved is often to maximize the rate of one user, say R_1, subject to minimum constraints for all other users, $R_n \geq c_n, n = 2, \ldots, N$. Thus, the complete algorithm includes an additional outer loop responsible for adjusting the weights w_n in such a way that the rate constraints are met. In general, the set of feasible rates is not known a priori, and so further experimentation may be required to determine appropriate rate constraints.

Performance Improvements of DSM Level 2 Algorithms

In this section we describe techniques for improving the speed, performance, and effectiveness of level 2 DSM. We first present a technique for increasing the speed of ISB by exploiting the concave-convex nature of the objective. Next, we introduce a utility-based approach to DSM that reduces computational complexity and improves the usability of algorithms. In particular, the utility formulation naturally discovers trade-offs between user rates without requiring a cumbersome exploration of the rate region via explicit rate constraints. We describe two approaches that we have developed for solving the utility-based DSM problem. In the first approach, we optimize the weighted rate sum with given weights (using ISB or SCALE) and then update the weights according to the utility function derivatives, iterating until convergence. In the second approach, we use a form of gradient ascent to directly optimize the utility objective.

Improved One-Dimensional Search

The innermost loop of the ISB algorithm is a one-dimensional optimization in which the Lagrange-discounted sum of rates on tone k, given by $\sum_{m=1}^{N} w_m R_m^k(P_n^k) - \lambda P_n^k$, is maximized as a function of the power of user n, with the powers of all other users held fixed. In general, this function is not concave and can have multiple local maxima, and for this reason the original definition of ISB used exhaustive search over a discrete set of power levels to perform the one-dimensional optimization. In contrast, we propose efficient global maximization methods that exploit the concave and convex nature of individual terms and do not require exhaustive search.

Using (1), the objective for a given tone and user can be expressed in the form

$$h(P) = w_n \log(1 + AP) + \sum_{m \neq n} w_m \log\left(1 + \frac{B_m}{C_m P + D_m}\right) - \lambda P$$

where A, B_m, C_m, D_m are constants determined by the powers P_l^k, background noise, and the direct and crosstalk gains. The first logarithmic term (the effect of a user's power on its own rate) is concave in P, while the remaining logarithmic terms (the effects of a user's power on the rates of other users) are convex in P. The objective can thus be expressed as the sum

$$h(P) = f(P) + g(P)$$

of a concave function f and convex function g (or equivalently, as the difference of concave functions) where the linear term λP can be included in either term.

Hat Bounds

Global optimization of the sum of a concave and convex function on an interval may be accomplished using upper bounds that we refer to as hat bounds. Consider a finite interval $[x_0, x_1]$. The convex function is upper-bounded on this interval by the secant of the graph, that is, by

$$g(x) \leq \frac{x_1 - x}{x_1 - x_0} g(x_0) + \frac{x - x_0}{x_1 - x_0} g(x_1)$$

The concave function f is upper bounded in the same interval by the tangents to the graph of f at the endpoints of the interval, that is, by the equations

$$f(x) \leq f(x_0) + f'(x_0)(x - x_0),$$
$$f(x) \leq f(x_1) + f'(x_1)(x - x_1).$$

Summing the bound on g with the minimum of the two bounds on f, one obtains a hat-shaped upper bound on the function h over the interval. The peak of the hat occurs at a point in the interval where the two bounds on f coincide.

A global optimization algorithm based on the hat bounds works as follows.

1. Construct the initial "hat" for the entire interval $[x_0, x_1]$.
2. Given a set of intervals with corresponding hat bounds, choose the interval $[a, b]$ whose hat bound has the highest peak.
3. Split the interval at the abscissa c where the peak occurs, forming two new intervals $[a, c]$ and $[c, b]$. Compute hat bounds for the new intervals.
4. Repeat from step 2 until the highest hat bound peak (an upper bound on the optimal value) is close to the highest measured point (a lower bound on the optimal value).

This simple method enables us to obtain the location of the optimum and its value within prescribed tolerances. **Figure 2** graphically depicts the first few iterations of the algorithm applied in one example function. The bottom curve depicts the (unknown) function h, which can be seen to have at least two local maxima in the interval. The solid, piecewise linear curve shows the set of four hat bounds obtained after three iterations of the algorithm, while the dashed lines depict earlier hat bounds that have been superseded. In the first iteration ($t = 1$), a single hat bound covered the entire interval. In the second iteration ($t = 2$), the interval was split to create two hat bounds. The right-most hat bound, having a higher peak than the bound on the left, was further split in the third iteration ($t = 3$). The figure also has five circles representing the points at which the function h was evaluated during the four iterations depicted.

If we change the value of λ in the definition of h, the hat bounds change linearly. This allows us to reuse hat bounds computed for one value of λ when redoing the optimization for a different value. Thus the inner loop of each user iteration (optimizing via hat bounds) becomes more efficient as the outer loop of that iteration (searching for the optimal Lagrange multiplier λ) proceeds. The outer loop of one user iteration searches for the right value of the Lagrange multiplier λ such that the spectrum of user n does not exceed its power constraint. The outermost loop cycles through iterations for all users until convergence is obtained.

Utility-Based Algorithms

As mentioned above, many DSM optimization methods require one to specify $N - 1$ rate constraints,

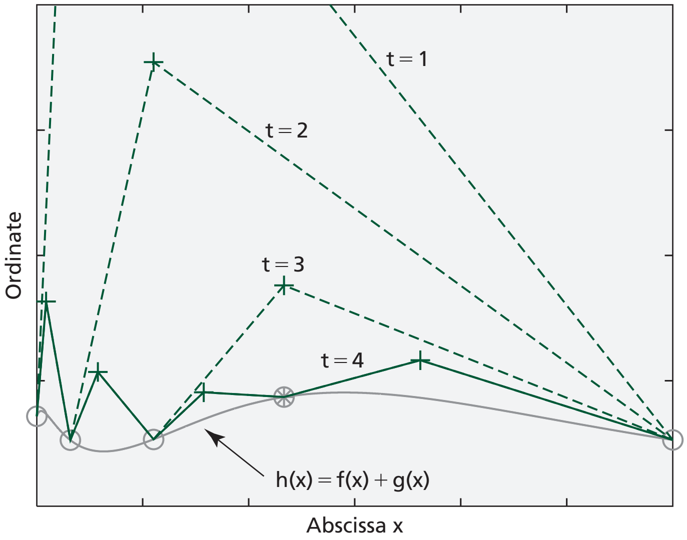


Figure 2.
Iterations of a global optimization algorithm.

with the objective being to maximize the *N* -th rate. However, the set of achievable rates is usually unknown so that it may be unclear as to whether a particular set of rate constraints is feasible or not. In practice, this leads to a trial and error approach of first trying one set of rate constraints and then another set, and so on, until a satisfactory solution is found. This affects the convergence speed, efficiency, and robustness of a spectral balancing algorithm. The resulting algorithms have many layers of iteration, with one layer designed to solve the problem for fixed weights, another designed to adapt the weights for given rate constraints, and yet another designed to find appropriate rate constraints.

Such difficulties can be avoided by expressing performance in terms of utility functions, functions of the rates, instead of the rates themselves. In particular, a utility function *U(R)* can be used to quantify the benefit that the system (or the user) obtains from a given transmission rate *R*. The optimization objective is to maximize the total utility, which is the sum of the individual utility terms. For example, one utility function might be based on a user's perception of quality of service. As the rate of an Internet connection goes up from tens of kilobits per second to hundreds of kilobits per second, a Web surfer notices significant gains in responsiveness. As the rate continues to increase into megabits per second, the gains become less noticeable and eventually saturate at the point where performance becomes limited by other bottlenecks in the network. Like most utility functions that arise in practice, such a function increases with increasing rates but exhibits a law of diminishing returns, meaning that the rate of increase *U'(R)* decreases with increasing rates. As another example, if the primary goal is to ensure a certain minimum rate to all users, one could use a utility function that increases sharply from zero to the minimum rate, and thereafter increases slowly. Optimization of a sum of such utilities is roughly equivalent to maximizing the sum rate subject to minimum rate constraints. Some advantages of a utility-based approach are:

- A priori information about the possible trade-offs among user rates is not required.
- Feasibility problems can be solved with graceful failure; if targets are not feasible, the utility approach will search for a point that is in some sense as close to feasible as possible.
- Information about the marginal benefit of increasing a user's rate, say from an economic or perceptual point of view, can be explicitly modeled via the utility function.
- A wide number of problems can be solved within the same optimization framework, merely by redefining the utility function.

- Utility-based optimization problems can be solved fairly quickly and mesh well with algorithms based on weighted rate sums.

Utility optimization using weighted rate sum algorithms. We now describe how a weighted rate sum optimizer, such as ISB or SCALE, can be leveraged in order to solve a utility-based objective function. In general, a utility function U provides a scalar value measuring the desirability of a rate vector **R**. Here, we will work with utility functions that are concave, continuously differentiable, and expressed as a sum of scalar utility functions for each user, although this last requirement is not necessary. The convexity of the utility function models the law of diminishing returns, in which the value of transmission at 2 Mbps is typically not more than twice the value of a rate of 1 Mbps. In the utility formulation, the problem is to maximize the objective function

$$J(\mathbf{R}) = \sum_{n=1}^{N} U_n(R_n)$$

over **R** in the feasible rate region corresponding to the given power constraints. Hence the problem is to maximize $J(\mathbf{R}(\mathbf{P}))$ subject to constraints on the power matrix **P**.

Recall that P_n^k denotes the power allocated to user n on tone k; then there are individual constraints $0 \le P_n^k \le \hat{P}_n^k$ on the power allocated to each tone as well as sum constraints on each line:

$$\sum_k P_n^k \le \overline{P}_n \ .$$

We choose the individual user utility functions to be non-negative, monotonically increasing, and concave. Non-negativity is not very important, although it is useful if one would like to use relative improvement as a criterion for halting iterations of an optimization algorithm.

Suppose that we have obtained a power spectrum $\boldsymbol{P}^*$ which optimizes $J(\mathbf{R}(\mathbf{P}))$. By Taylor's theorem, J can be approximated locally by a weighted sum of rates, where the weights are the derivatives $U'_n(R_n^*)$ of the utility functions at the optimal rates $\boldsymbol{R}^* = \boldsymbol{R}(\boldsymbol{P}^*)$. Thus, there is a set of weights w such that $\boldsymbol{P}^*$ is at least a local optimizer for $\sum w_n R_n(\mathbf{P})$. Moreover, when the number of tones is large, the rate region is known to be approximately convex. When the rate region is convex, the statement can be strengthened to say that **P*** is a global optimizer for a certain weighted rate sum. Thus, given the correct set of weights, optimizing the utility and optimizing the weighted sum solve the same problem. Unfortunately, due to the nonconvexity of the rate function as a function of power, the weighted rate sum may have local optima as well.

For similar utility optimization problems in wireless systems, there is a well-known method for iteratively adapting weights in order to maximize utility, often in a stochastic context. When the utility function is the logarithm of the rate, this method is referred to as the proportional fair algorithm [10]. In this method, one defines a throughput vector T, which is an exponentially smoothed average of the rate vector R obtained in past iterations. In each iteration, one chooses weights based on local approximation of the utility function in the neighborhood of the current throughput. The rate for the next iteration is chosen by maximizing the weighted rate sum. This algorithm has proven convergence properties in a number of settings, when the exponential smoothing has sufficiently long memory. The following algorithm is a natural way to solve our utility maximization problem using the proportional fair approach.

1. Initialization: Choose an initial power spectrum $\mathbf{P}(0)$, set time $t = 0$, and let the initial throughput be $\mathbf{T}[-1] = \mathbf{R}(\mathbf{P}[0])$.
2. Update the throughput vector as $\mathbf{T}[\mathrm{t}] = (1-\varepsilon)$ $\mathbf{T}[\mathrm{t} - 1] + \varepsilon\mathbf{R}(\mathbf{P}[\mathrm{t}])$.
3. Set weight vector $\mathbf{w}[\mathrm{t}]$ as $w_i[t] = U_i'(T_i[t])$.
4. Set $\mathbf{P}$ $[t + 1]$ to be the argmax of $g_i w_i[t]R_i(\boldsymbol{P})$ subject to power constraints, computed using ISB with initial spectrum $\mathbf{P}[t]$.
5. Repeat step 2 through step 4 until $\mathbf{R}(\mathbf{P}[\mathrm{t} + 1])$ is approximately equal to $\mathbf{T}[\mathrm{t}]$.

When the smoothing factor ε is small enough, the algorithm is expected to converge, in which case the power spectrum is a local optimizer of the utility function.

Optimizing utility by constrained gradient ascent. In the absence of power constraints, a natural approach to maximizing the utility function $J(\mathbf{R})$ would be to use direct, gradient-based hill climbing. Given a power spectrum $\mathbf{P}(t)$, it is straightforward to compute the gradient $\boldsymbol{v}$ of J with respect to $\mathbf{P}$. In principle, one could always improve the utility by gradient ascent. That is, one does a one-dimensional maximization of $J(\mathbf{R}(\mathbf{P}[\mathrm{t}] + \alpha\, \boldsymbol{v}[\mathrm{t}]))$over the scalar α to obtain $\mathbf{P}[\mathrm{t} + 1]$. By repeating this, one obtains a sequence of power spectra with increasing utility, converging to a local maximum. The main obstacle to implementing this approach is the presence of power constraints: for a power vector $\mathbf{P}[\mathrm{t}]$ that meets some constraints with equality, the vector $\mathbf{P}[\mathrm{t}] + \alpha\boldsymbol{v}[\mathrm{t}]$ may not satisfy the constraints for any positive value of α. Indeed, line-search methods are normally used for unconstrained optimization. As discussed below, line search methods can in principle be extended to constrained optimization problems by using a projection operator that maps an unconstrained space into a constrained space. However, constrained gradient ascent is often impractical, since the computation of the projection operator itself may be computationally intensive. Fortunately, because the power constraints in our problem have a simple structure, the projection operator can be computed efficiently, and we have found that constrained gradient ascent is a computationally efficient way to find at least a local maximum of the total utility objective.

The constrained gradient optimization method described here has certain advantages. It is easy to implement and in many cases converges quickly to a good solution. It can be applied to "tweak" the result of a solution obtained by any other method, as it always strictly improves the utility in each step. On the downside, if it is given a poor starting point, then the method is vulnerable to being trapped in a local maximum. We have found this method to be very effective in conjunction with iterative waterfilling. Iterative waterfilling is applied as a starting point, and constrained gradient optimization is then applied to improve the utility of the solution. Another technique for avoiding undesirable local minima is to use an annealing approach. Here, we initially start with no crosstalk and gradually increase the level of crosstalk while continuously re-optimizing via gradient ascent.

We have found that the method scales very well with respect to the number of lines and tones. The complexity of each individual line search increases quadratically with the number of lines N, and linearly with the number of tones K. The dependence of the number of line searches required as a function of N and K is not clear; however, one tends to get the largest gains in the earlier iterations, so that one can generally obtain improvement, if not optimization, in a relatively small number of iterations. In the following text we provide some details on the constrained gradient optimization method we employed, before moving on to some numerical examples.

The key idea of projection-based constrained optimization is to define a projection Π that maps each point in an unconstrained space (say the m-dimensional Euclidean space $\mathcal{R}^m$) to the subset C of the space satisfying the constraints. Then, in principle, one can replace the problem of maximizing $f(\boldsymbol{x})$ over C with the unconstrained problem of maximizing $f(\Pi(\mathbf{y}))$ over $\boldsymbol{y}$ in $\mathcal{R}^m$. A general algorithm for line search based optimization is:

1. Take a starting point $\boldsymbol{x}[0] \in C$ and set $n = 1$
2. Choose a search direction $\boldsymbol{v}[n]$, for example, letting $\boldsymbol{v}[n]$ be the gradient of f at $\boldsymbol{x}[n - 1]$
3. Perform line search: find t^* to maximize $f(\Pi(\mathbf{x}[n - 1] + t\mathbf{v}[n]))$
4. Set $\mathbf{x}[n] = \Pi(\mathbf{x}[n - 1] + t * \mathbf{v}[n])$
5. Repeat step 2 through step 4 until convergence

In particular, we used the nearest neighbor projection

$$\Pi(\mathbf{y}) = \underset{\mathbf{x} \in C}{\arg\min} \| \mathbf{x}-\mathbf{y} \|$$

which is well-defined when C is a convex set. In our case, C is a convex polytope defined by the power constraints. Due to the simple nature of the power constraints, the projection operator may be computed efficiently. For a single line with sum constraint $\overline{P}$, and with maximum power constraint $\hat{P}_k$ for tone $k = 1, \ldots,$ K, the projection operator may be defined by the following algorithm:

1. Given $P_k, k = 1, 2, \ldots ,K$
2. Set P_k to $\max(P_k, 0)$ then to $\min\,(P_k, \hat{P}_k)$
3. If $\sum_k P_k > \overline{P}$ and if m elements satisfy $P_k > 0$, subtract $\frac{\Sigma_k P_k - \overline{P}}{m}$ from each element satisfying $P_k > 0$
4. Repeat step 2 through step 3 until convergence.

The projection operation is guaranteed to converge in no more than K iterations. The complete projection operator Π for multiple lines is defined by applying the above algorithm to each line independently.

There are a number of computational issues that should be considered in implementing this algorithm that are beyond the scope of this paper. With these issues properly accounted for, we found this method to be significantly faster than methods based on adapting weights.

Numerical Results

To illustrate the results that can be obtained with the algorithms described above, we performed some numerical simulations of DSL systems. The system parameters are generally based on asymmetric digital subscriber line (ADSL) downstream scenarios used in [3, 4]. The line diameters are 0.5 mm, tone spacing is 4.3125 kHz, and the symbol rate is 4 kHz. The power on each modem is constrained to 20.4 dBm and the noise model is ANSI A, with a signal-to-noise ratio (SNR) gap of 12.9 dB.

Two-line, CO/RT scenario. The need for spectral optimization and the role of utility functions are most simply illustrated in a two-line near-far scenario. Line 1 (CO) is deployed from the central office and is 5 km long, while line 2 (RT) is deployed from a remote terminal located 4 km from the CO and is 3 km long. Due to the near-far effect, line 2 can cause significant interference to line 1 but receives little interference in return. The dashed curve in **Figure 3** depicts the rate region obtained using the IWF algorithm, while the solid curve depicts the much larger rate region obtained using the utility maximization (UM) algorithm (using linear weights as utilities). Because of the length of the CO line, it can best use lower frequencies. As reported in [5], the OSB algorithm reserves a certain number of the lowest tones for the CO line, while allowing the RT to use the higher tones. Iterative waterfilling gives poor rates to the CO because the RT spectrum places equal emphasis on low and high tones.

When using utility maximization, different choices for utility functions will lead to different points on the

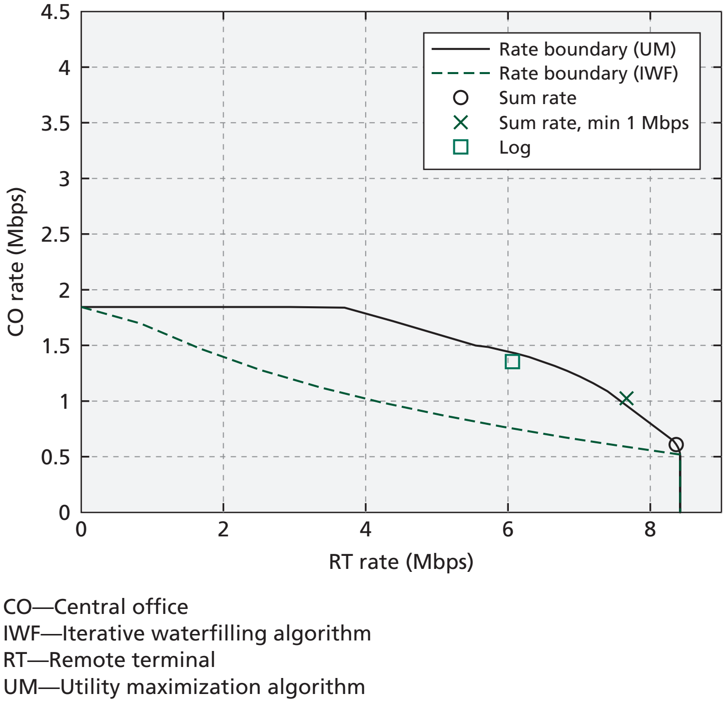


Figure 3.
Rate regions obtained using the IWF and UM algorithms.

rate region boundary. For example, if the utility J(**R**) is defined to simply be the sum of the rates, then UM finds the rate pair marked by a circle. In another case, we used smoothed piecewise linear utilities, where the slope of the utility function was approximately 100 for rates below 1 Mbps and approximately 1 for rates above 1 Mbps. In other words, the first priority is to achieve 1 Mbps for all users, and the second priority is to maximize the sum rate. In this case, the UM algorithm obtained the point marked by an x, effectively the largest sum rate possible subject to both users achieving 1 Mbps. Finally, the square mark shows the rates obtained using $U_n(R_n) = \log(1 + 500R_n)$. The logarithmic utility represents a generic law of diminishing returns that tends to seek a balance between the goals of high sum rate and fairness between lines.

Four line scenario. We next present results for a four-line scenario. As in [4], the lines are located at distances 0, 2, 3, and 4 km from the CO and have lengths 5, 4, 3, and 2 km, respectively. It is not feasible to graphically depict the entire rate region in this case, but in **Figure 4** we show the rates achieved on each line, with different utility functions. The upper curve in the figure shows the maximum rate achievable by each line in the absence of crosstalk. The curve marked by circles shows the rates obtained when maximizing the sum of rates. Here priority is given to line 4 at the expense of line 1. The curve marked by x shows the result of using the approximately piecewise linear utility functions, described in the two-line scenario above, to enforce a minimum rate of 1 Mbps for each line. Here the rates of lines 4 and 3 are reduced in order to bring line 1 and line 2 up to the minimum rate. Finally, the curve marked by squares depicts the rates achieved using identical logarithmic utilities for each user.

Twenty-four-line scenario with random crosstalk amplitudes. To illustrate the scalability of the utility optimization method, we present results for a 24-line bundle. In the previous two scenarios, we used a crosstalk model in which the amplitude is intended to represent the 99th percentile of the crosstalk observed in practice. In this 24-line scenario, we used a model with random crosstalk realizations drawn from a distribution. In particular, we modulated the worst case crosstalk amplitudes by log normal random variables, drawn from a distribution with the

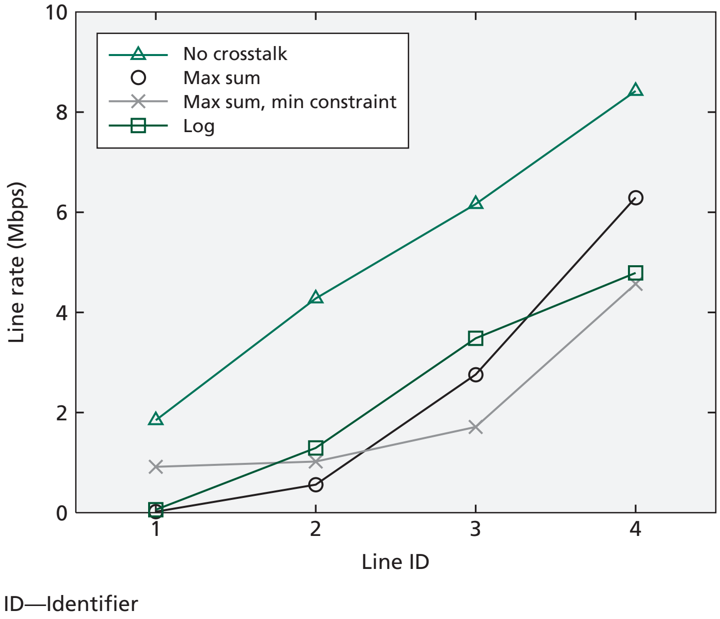
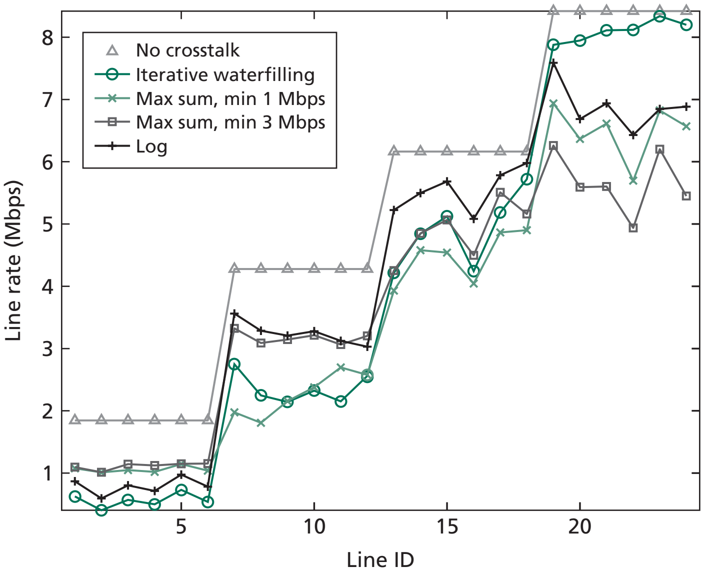


Figure 4.
Rates achieved on each line, with different utility functions.

99th percentile equal to that predicted by the worst case model. The standard deviation of the log of the log normal random variables was 7 dB. Six lines have the same line length and starting position as line 1 in the four-line scenario, six match line 2 from that scenario, and so on. The rate results are depicted in **Figure 5**. The graph with the tag by "no crosstalk" presents the rates obtainable in the absence of crosstalk, which follows a step function since it depends on line length. The curve marked by circles shows the rates achieved by iterative waterfilling (with maximum power used on each line). Note that the fourth group of lines gets excellent rates but that the first group performs very poorly in this case. The rates marked with x show the result of using approximately piecewise linear utility functions with very high slope below 1 Mbps. The rate for the first group of users increases to this minimum rate, while the rate for some other users (notably belonging to the fourth group) decreases to accommodate this improvement. When the high slope region is extended to 3 Mbps to attempt to enforce a minimum rate of 3 Mbps, the resulting rates are marked by squares. The second group is brought up to the target level, largely through further sacrifices of rates in the fourth group. The first group is limited to just above 1 Mbps by intra-group interference and interference from the second group. Finally, the remaining curve shows the rates obtained using log utilities. Taken together, these examples illustrate the ability of utility maximization to determine reasonable operating points on the rate region boundary, without needing explicitly to determine or explore the entire, high-dimensional rate region. The 3 Mbps example also illustrates the "soft failure" nature of the utility approach, which tends to do as well as possible, even when the implied goal is not feasible.

Solution for Deployment of DSM Spectrum Optimization Algorithms

In this section we describe a solution for in-field deployment of our algorithms. Different system-level aspects and requirements for this solution were taken into account during its design. The key component of our solution, the DSM Center, is independent of the legacy equipment deployed in the field, as long as this



ID—Identifier

Figure 5.
Rate results.

equipment conforms to the standards. We show the architecture of the DSM Center, briefly analyze its implementation and indicate future steps required to convert it to a product.

System Architecture

As already discussed, there are a plethora of DSM spectrum optimization solutions—some of these use centralized algorithms, while the others use distributed algorithms. In general, one may say that the (near)-optimal algorithms suffer from high complexity, while the fast algorithms are suboptimal. Some of the algorithms use target rates, while for utility-based algorithms these may not be necessary. If one DSLAM controls multiple binders, there may be a need to run different algorithms for different binders, so there is a need to deploy multiple algorithms per DSLAM. The choice of algorithm used for a binder is made before the startup of the system. For our experiments, we deployed our solution (DSM Center) on a personal computer (PC) platform that communicates with the DSLAM(s) using the Simple Network Management Protocol (SNMP).

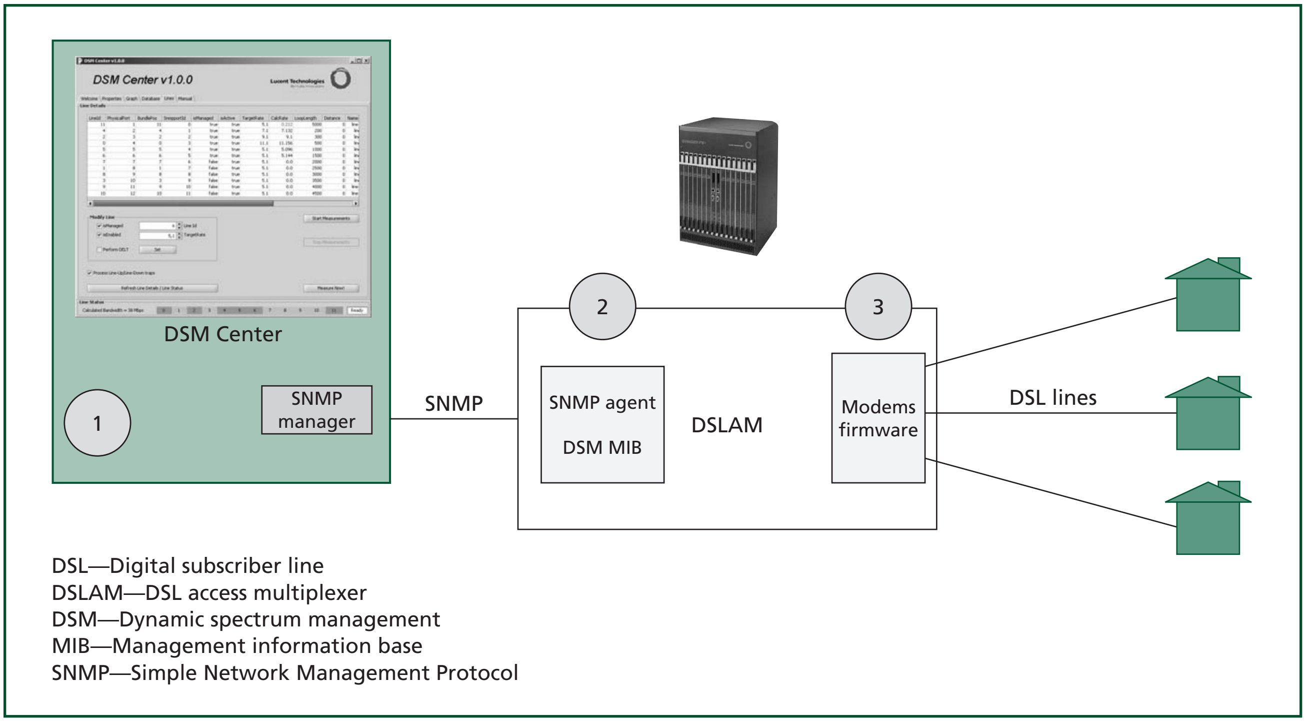
The DSM Center collects data from the DSLAM, such as transmit power per bin, receive power per bin, bit loading, rates, and SNR margin. Chipset vendors need to support these mechanisms. The crosstalk channel could be obtained using different channel estimation algorithms (these were not implemented) or measured beforehand and stored in the database, which is the solution we used for the DSM Center. Once these parameters have been collected at the DSM Center, new transmission parameters are calculated by a DSM spectrum optimization algorithm. These parameters are sent via SNMP to the DSLAM, and the last step of the solution is to adjust these parameters "on the fly" without retraining and without bit errors. Typical transmission parameters that should be adjusted in this way are the PSD mask and the SNR margin. Seamless adjustment of the parameters, however, is not yet supported and will require further development in cooperation with chipset vendors. **Figure 6** illustrates the system architecture of the DSM spectra optimization solution. There are three major components:

1. The DSM Center, where information about the lines is monitored and/or changed, and where new power spectra are determined by running DSM algorithms.
2. The DSM-related management information base (MIB) implementation at the DSLAM, including the SNMP agent.
3. The modem firmware that implements the DSM control function.

We have implemented a prototype system including the first two components, and have specified requirements for the modem firmware.

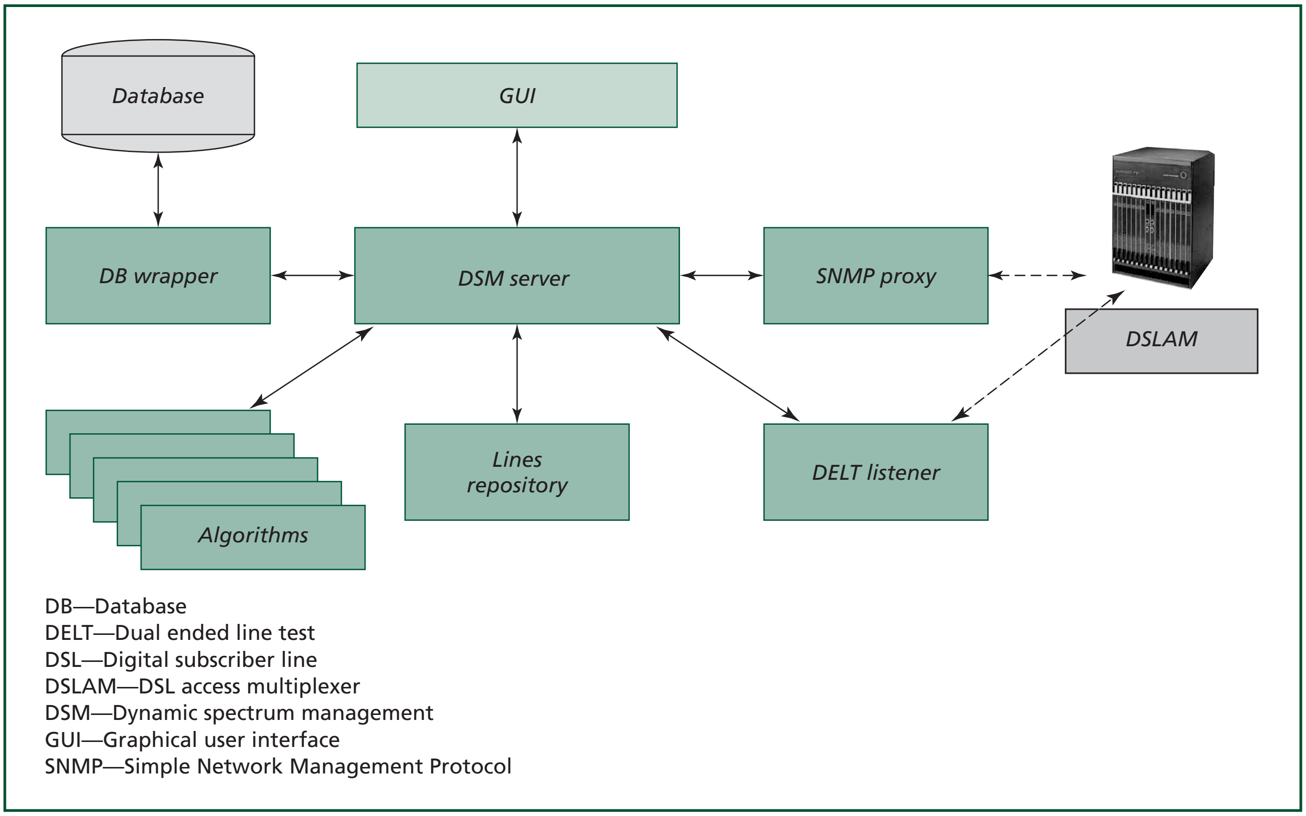
DSM Center Architecture

The architecture of the DSM Center is shown in **Figure 7**. Using the factory design pattern [7], multiple algorithms could be supported. Only one of the algorithms could be used for a given binder, and the algorithm used is specified in a configuration file before the DSM Center is started. This parameter is read by the DSM Center at startup and a particular algorithm instance is created. In order to provide the DSM Center with basic information about the lines, a database is used, and the access to the database is implemented using the wrapper design pattern [7]. The database contains, for example, the port-binder-algorithm identification mapping for any particular line that is controlled by the solution. The data read from the database as well as the control parameters per line (e.g., SNR or margin) are stored in the repository. The monitoring of the system, visualization of the PSD masks calculated for a selected algorithm, as well as change of the parameters are provided by a graphical user interface (GUI), shown in **Figure 8**. The GUI is implemented using the model-view-control (MVC) design pattern [7]. The SNMP communication with the DSLAM takes place via the SNMP proxy, which has been implemented according to the proxy design pattern [7]. There is a timeout per SNMP request established. Once timeout expires without receiving an SNMP response, the DSM Center reissues the request, and after the maximum number of unsuccessful tries is reached, a warning is issued to the end user. The diagnostics of the particular line as well as gathering of additional line information can be done using dual ended line testing (DELT). Once the DELT command is issued to the DSLAM using the GUI, DELT begins. The DELT listener running in



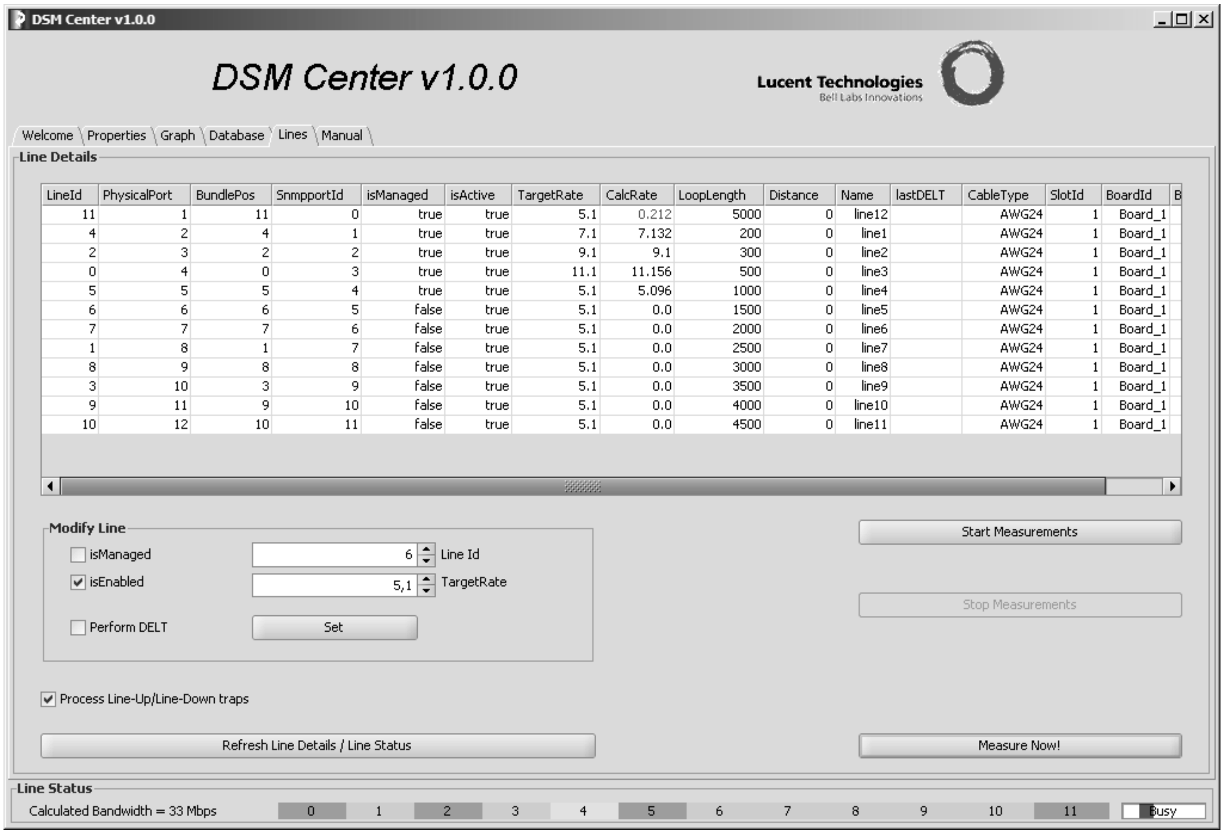
DSL—Digital subscriber line
DSLAM—DSL access multiplexer
DSM—Dynamic spectrum management
MIB—Management information base
SNMP—Simple Network Management Protocol

Figure 6.
The system architecture of the DSM spectra optimization solution.



DB—Database
DELT—Dual ended line test
DSL—Digital subscriber line
DSLAM—DSL access multiplexer
DSM—Dynamic spectrum management
GUI—Graphical user interface
SNMP—Simple Network Management Protocol

Figure 7.
Architecture of the DSM Center.



Line Details

LineId	PhysicalPort	BundlePos	SnmpportId	isManaged	isActive	TargetRate	CalcRate	LoopLength	Distance	Name	lastDELT	CableType	SlotId	BoardId	B
11	1	11	0	true	true	5.1	0.212	5000	0	line12		AWG24	1	Board_1	
4	2	4	1	true	true	7.1	7.132	200	0	line1		AWG24	1	Board_1	
2	3	2	2	true	true	9.1	9.1	300	0	line2		AWG24	1	Board_1	
0	4	0	3	true	true	11.1	11.156	500	0	line3		AWG24	1	Board_1	
5	5	5	4	true	true	5.1	5.096	1000	0	line4		AWG24	1	Board_1	
6	6	6	5	false	true	5.1	0.0	1500	0	line5		AWG24	1	Board_1	
7	7	7	6	false	true	5.1	0.0	2000	0	line6		AWG24	1	Board_1	
1	8	1	7	false	true	5.1	0.0	2500	0	line7		AWG24	1	Board_1	
8	9	8	8	false	true	5.1	0.0	3000	0	line8		AWG24	1	Board_1	
3	10	3	9	false	true	5.1	0.0	3500	0	line9		AWG24	1	Board_1	
9	11	9	10	false	true	5.1	0.0	4000	0	line10		AWG24	1	Board_1	
10	12	10	11	false	true	5.1	0.0	4500	0	line11		AWG24	1	Board_1	

Modify Line

☐ isManaged — 6 Line Id

☑ isEnabled — 5,1 TargetRate

☐ Perform DELT

☑ Process Line-Up/Line-Down traps

Line Status

Calculated Bandwidth = 33 Mbps — 0 1 2 3 4 5 6 7 8 9 10 11 — Busy

DSM—Dynamic spectrum management

Figure 8.
DSM Center graphical user interface.

the background is notified once the file with the results of DELT are downloaded (using File Transfer Protocol [FTP] and a known port) from the DSLAM to the PC where the DSM Center is running.

Before the DSM Center is started, a few parameters need to be specified:

1. The DSM algorithm to be used,
2. The FTP server's Internet Protocol (IP) and port address, i.e., the IP address of the PC where the DSM Center is located,
3. IP address of the DSLAM,
4. SNMP time-out value, and
5. Control information of the database.

Once the system starts, it retrieves relevant data from the database and populates the repository with information about the lines. The information read is used to update the GUI. From the GUI, it is possible to select which lines in the binder will be controlled by the algorithm, and which will only be monitored. The selected algorithm is instantiated and the periodic measure-calculate-write cycle is started. This cycle can have a period ranging from a couple of seconds up to several hours. Periodic execution could be stopped (e.g., to change the target bit rates) and restarted via the GUI. In addition, it is possible to "asynchronously" request the execution of the given algorithm at any instance of time using the GUI.

The measurements (and parameters to be measured) are requested using the SNMP protocol. There is a timeout logic for completion of the requests, and once a maximum number of unsuccessful requests has been made, the algorithm stops execution and a warning is issued. In the case where a successful request is made, the collected data is sent to the DSM

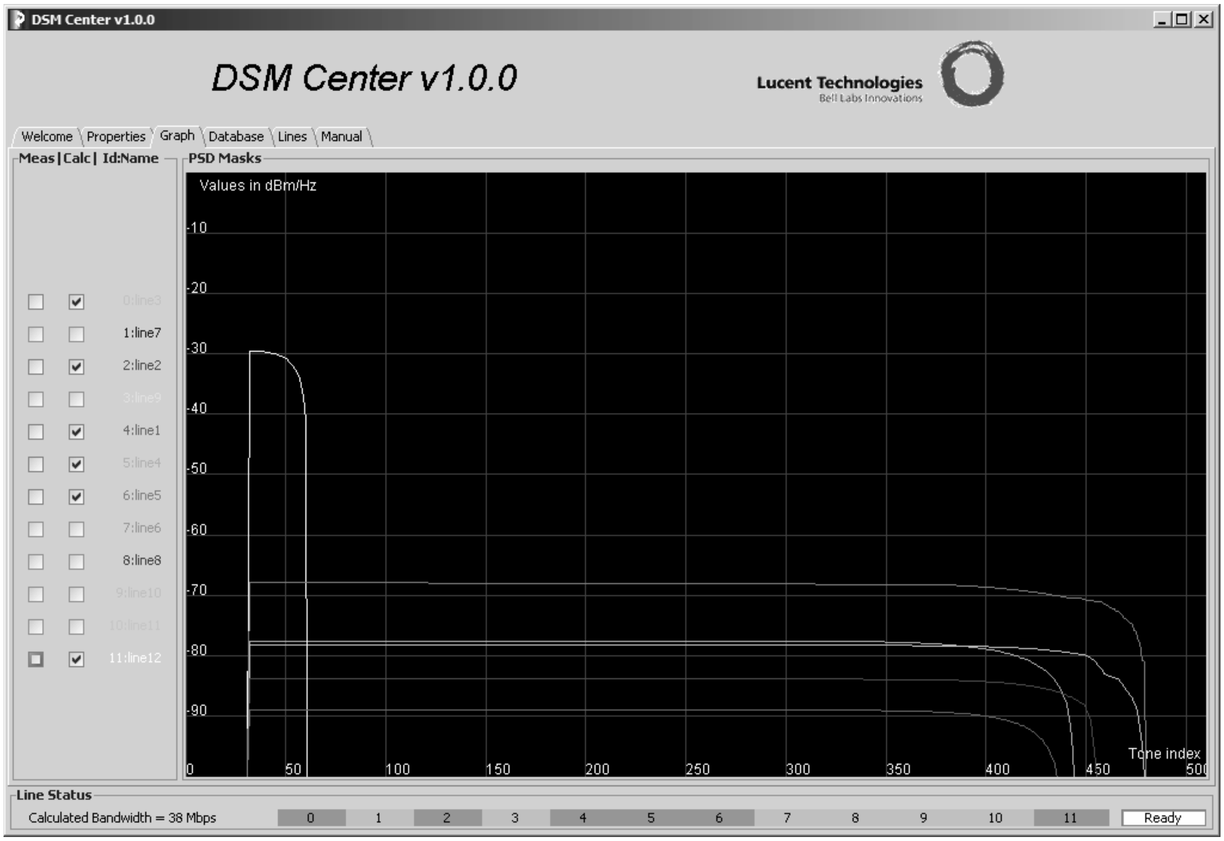
Center and the algorithm is executed. The results of the algorithm (such as updated PSD masks) could be sent to the DSLAM when these differ from the values calculated in the previous cycle. An example of updated PSD masks represented at the GUI is given in **Figure 9**.

When a line in the binder is activated (called a LineUp event) or a line in the binder is de-activated (called a LineDown event), a new calculation cycle is executed and new values are provided by the DSM Center. It may happen that a particular line shows instability, manifested in a number of consecutive LineUp and LineDown events. In such a case, it is possible either to exclude the troubled line from the calculations (specifying that the line will be monitored only, and not controlled) or to specify that SNMP traps sent by the DSLAM for a particular line be ignored.

The firmware of the system has not yet been developed. We therefore executed the entire algorithm cycle without applying the PSD masks at the CPE modems. The following has been verified

- The performance results of the algorithm are close to those expected.
- There is SNMP communication between the DSLAM platform and the DSM Center, which ensures that the parameters currently implemented at the modem's firmware can be obtained and that the algorithm results are obtained by the DSLAM.

In order to verify point 1, we compared the algorithm output with the simulations of the same



Meas	Calc	Id:Name
□	☑	0:line3
□	□	1:line7
□	☑	2:line2
□	□	3:line9
□	☑	4:line1
□	☑	5:line4
□	☑	6:line5
□	□	7:line6
□	□	8:line8
□	□	9:line10
□	□	10:line11
▣	☑	11:line12

Calculated Bandwidth = 38 Mbps

DSM—Dynamic spectrum management
PSD—Power spectrum density

Figure 9.
Updated PSD masks.

algorithm performed in MATLAB*. There was a good match between the two.

One capability that the DSLAM still lacks is the ability to instruct all controlled modems to apply newly calculated masks simultaneously. Note that the PSD masks are sent to the DSLAM in a consecutive manner via SNMP. Therefore, the DSLAM needs to be able to obtain and store new masks for all users before they are applied. This is not difficult in principle but will require the cooperation of chipset vendors.

Conclusions and Further Perspectives

In this paper, we have discussed improvements of DSM level 2 algorithms that are easy to implement, lead to performance improvements of DSL, and are computationally more efficient than previously developed algorithms. With these improvements, it is feasible to optimize transmission parameters (like PSD masks) of all lines in a binder in real time. This capability is a cornerstone of our solution for unified and easy deployment of DSM spectrum optimization algorithms in the field. The solution is based on the development of a new component, the DSM Center, that could be provided as an independent network element or could be integrated as part of the DSLAM. We compared the results from the platform with the simulations and found a good match between the two with respect to speed of execution and other calculated parameters. The next step needed to implement a full solution for DSM spectrum optimization algorithms in the field is an update of modem firmware in cooperation with chipset vendors.

Acknowledgements

The authors would like to thank the following people for their contributions to the paper: Philippe Hervé, Mohamed Najar, Willem Romijn, Mans van Tellingen, and Marcus Weldon.

*Trademarks

MATLAB is a registered trademark of The Mathworks, Inc.

References

[1] Alliance for Telecommunications Industry Solutions, Network Interface, Power and Protection Committee (NIPP), Network Access Interfaces Subcommittee (NAI), Draft Dynamic Spectrum Management Tech. Report, NIPP-NAI-2007-038R3 Draft TR, May 2007.

[2] American National Standards Institute, “Spectrum Management for Loop Transmission Systems,” ANSI T1.417-2003.

[3] R. Cendrillon, Multi-User Signal and Spectra Co-ordination for Digital Subscriber Lines, Ph.D. Thesis, Catholic University Leuven, Dec. 2004.

[4] R. Cendrillon and M. Moonen, “Iterative Spectrum Balancing for Digital Subscriber Lines,” Proc. IEEE Internat. Conf. on Commun. (ICC ‘05) (Seoul, Korea, 2005), vol. 3, pp. 1937–1941.

[5] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, “Optimal Multi-User Spectrum Management for Digital Subscriber Lines,” Proc. IEEE Internat. Conf. on Commun. (ICC ‘04) (Paris, Fr., 2004), vol. 1, pp. 1–5.

[6] S. Galli and K. Kerpez, “Methods of Summing Crosstalk from Mixed Sources—Part I: Theoretical Analysis,” IEEE Trans. Commun., 50:3 (2002), 453–461.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, Reading, MA, 1994.

[8] J. Huang, R. Cendrillon, M. Chiang, and M. Moonen, “Autonomous Spectrum Balancing (ASB) for Frequency Selective Interference Channels,” Proc. IEEE Internat. Symposium on Inform. Theory (ISIT ‘06) (Seattle, WA, 2006), pp. 610–614.

[9] K. Kerpez and S. Galli, “Methods of Summing Crosstalk from Mixed Sources—Part II: Performance Results,” IEEE Trans. Commun., 50:4 (2002), 600–607.

[10] H. Kushner and P. Whiting, “Convergence of Proportional-Fair Sharing Algorithms Under General Conditions,” IEEE Transactions on Wireless Communications, 3:4, (July 2004), 1250–1259

[11] S. H. Lin, “Statistical Behavior of Multipair Crosstalk,” Bell Syst. Tech. J., 59:6 (1980), 955–974.

[12] J. Papandriopoulos and J. S. Evans, “Low-Complexity Distributed Algorithms for Spectrum Balancing in Multi-User DSL Networks,” Proc. IEEE Internat. Conf. on Commun. (ICC ‘06) (Istanbul, Tur., 2006), pp. 3270–3275.

[13] C. Valenti, “Cable Crosstalk Parameters and Models,” ANSI contribution T1E1.4/97-302, Telcordia Technologies, Sept. 1997.

[14] W. Yu, G. Ginis, and J. M. Cioffi, “Distributed Multiuser Power Control for Digital Subscriber Lines,” IEEE J. Select. Areas in Commun., 20:5 (2002), 1105–1115.

(Manuscript approved November 2007)

MIROSLAV ŽIVKOVIĆ is a member of technical staff in Alcatel-Lucent Bell Labs Europe in Hilversum, The Netherlands. He holds a Dipl. Ing. degree in electronics and telecommunications from the Faculty of Electrical Engineering at the University of Belgrade in Belgrade, Serbia. His current research interests are in the area of dynamic spectrum management for digital subscriber line (DSL) systems.

GERHARD KRAMER is a member of technical staff in Bell Labs’ Communications and Statistical Sciences Department in Murray Hill, New Jersey. He received B.Sc. and M.Sc. degrees in electrical engineering from the University of Manitoba, in Winnipeg, Canada, and the Dr. sc. techn. (Doktor der Technischen Wissenschaften) degree from the Swiss Federal Institute of Technology (ETH), in Zurich, Switzerland. Dr. Kramer is currently serving as an Associate Editor for Shannon Theory for the IEEE Transactions on Information Theory. He is a co-recipient of the IEEE Communications Society 2005 Stephen O. Rice Prize paper award for his work on coded modulation for multi-antenna communication.

CARL J. NUZMAN is a member of technical staff in the Complex Systems Analysis and Optimization Department at Bell Labs in Murray Hill, New Jersey. He received B.S. degrees in electrical engineering and mathematics from the University of Maryland at College Park, and a Ph.D. in electrical engineering from Princeton University in Princeton, New Jersey. Dr. Nuzman has broad research interests in the areas of optimization, optical networking, dynamical systems, and stochastic processes.

CARL POSTHUMA is a distinguished member of technical staff at Alcatel-Lucent in Naperville, Illinois. He is currently involved in digital subscriber line (DSL) standards activities and has been involved in voice quality standards in the past. He holds a B.S.E.E. and M.S.E.E. in electrical engineering from the University of Michigan in Ann Arbor. He has received 20 patents mainly in the areas of DSL, voice, and combined voice and DSL systems. His current interests include DSL, voice over IP, and other physical layer technologies.

JAMES WHEELER is a distinguished member of technical staff in Alcatel-Lucent’s Access Division in Whippany, New Jersey. He is currently product line manager for the Stinger® DSL Access Concentrator. He holds a B.Sc. in physics from the North Georgia College and State University in Daholenga, Georgia, and B.Sc. and M.Sc. in electrical engineering from the Georgia Institute of Technology in Atlanta. His interests include digital subscriber line (DSL) physical layer and carrier systems operations.

PHIL WHITING is a member of technical staff in the Complex System Analysis and Optimization department of Bell Labs in Murray Hill, New Jersey. He received his B.A. degree from the University of Oxford, his M.Sc. from the University of London, and his Ph.D. in queuing theory from the University of Strathclyde, all in the United Kingdom. After a post-doc at the University of Cambridge, his interests centered on wireless. His main interest is the mathematics of wireless networks, particularly stochastic models for resource allocation and information theory. His current research includes large deviation asymptotics for balls and urns models and the application of stochastic control to scheduling in wireless data networks

ADRIAAN J. DE LIND VAN WIJNGAARDEN is a member of technical staff in Bell Labs’ Communications and Statistical Sciences Department in Murray Hill, New Jersey. He received his M.S. in electrical engineering from Eindhoven University of Technology, The Netherlands, and his doctorate in engineering from the University of Essen, Germany. He has been deeply engaged in both theoretical and application-driven research in communications, information theory, coding, combinatorial and algorithmic optimization. He provided key contributions to Alcatel-Lucent’s 10G and 40G optical systems, in-building wireless and broadband access. He has authored more than 60 technical papers, received four patents, and has several patents pending. Dr. de Lind van Wijngaarden is a Senior Member of the IEEE and serves as a Publication Editor of the IEEE Transactions on Information Theory. He has co-organized Shannon Day at Bell Labs in 1998, Shannon’s Statue Unveiling at Bell Labs in 2001, DIMACS conferences, and several international workshops. ◆