Crosstalk Mitigation Techniques for Digital Subscriber Line Systems

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg

zur Erlangung des akademischen Grades

Doktor-Ingenieurin (Dr.-Ing.)

genehmigte Dissertation

von

Monique Düngen

aus

Moers

2016

1. Gutachter
Prof. Dr. rer. nat. Hermann Rohling
Technische Universität Hamburg-Harburg

2. Gutachter
Prof. Dr.-Ing. Udo Zölzer
Helmut Schmidt Universität

3. Gutachter
Prof. Dr. rer. nat. Volker Turau
Technische Universität Hamburg-Harburg

Tag der mündlichen Prüfung: 13. Januar 2016

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Nachrichtentechnik der Technischen Universität Hamburg-Harburg. Ein besonderer Dank gilt Herrn Prof. Rohling für die Betreuung meines Promotionsvorhabens, für die hilfreichen Diskussionen und Kommentare und dafür, dass er mir die Zeit gegeben hat, mit meiner Dissertation ins Reine zu kommen. Außerdem möchte ich ihm für den Blumenstrauß an Erfahrungen danken, den ich in der Zeit an seinem Institut in der Betreuung von Vorlesungen, der Arbeit mit Studenten und Kollegen, in Industrie- und Forschungsprojekten und auf Fachkonferenzen erhalten durfte. Herrn Prof. Zölzer danke ich für die Übernahme des Zweitgutachtens und das gute Gespräch nach meiner Doktorprüfung. Bei Herrn Prof. Turau möchte ich mich für das Drittgutachten und für seine interessanten Fragen in der Doktorprüfung bedanken. Alternative Denkansätze weiß ich sehr zu schätzen. Zudem geht mein Dank an Herrn Prof. Brinkmeyer, den Vorsitzenden des Prüfungsausschusses, für den professionellen und angenehmen Prüfungsablauf.

Den ehemaligen Kolleginnen und Kollegen bin ich für die tolle Arbeitsatmosphäre und die schöne Zeit am Institut dankbar. Neben der guten fachlichen Zusammenarbeit, denke ich auch gerne an die Weihnachtsfeiern, Karaokeabende, Institutsausflüge, Triathlons usw. zurück. Besonders möchte ich mich bei Dr. Rainer Grünheid, Alejandro Garcia Monsalve, Dr. Yajie Ruan und Dr. Steffen Heuel bedanken. Ihr habt mir durch eure Fachkompetenz, eure Abschlussarbeiten, als Korrekturleser, Kümmerer oder moralisches Standbein sehr geholfen.

Nicht zuletzt möchte ich mich bei meiner Familie und meinen guten Freunden bedanken. Ihr habt mich immer unterstützt und bestärkt. Nora, Ruby und Kurt danke ich für die Beschäftigung mit dieser speziellen Thematik beim Korrekturlesen. Bei Lina, Imke, Bahar, Karo und Julia möchte ich mich für die passende Ablenkung zur richtigen Zeit bedanken. Ihr habt dafür gesorgt, dass ich meine Doktorarbeit auch mal für eine Weile vergessen konnte. Ein ganz besonderes Dankeschön gilt Philip für seine Geduld und seinen Rückhalt während der langen Zeit des Schreibens. Danke dafür, dass Du auf viele gemeinsame Stunden verzichtet

hast und nicht müde geworden bist, meine Fragen ob ich das schaffe und jemals fertig werde, immer wieder zu bejahen.

Hannover, Februar 2016 Monique Düngen

Contents

1

Introduction

Digital communications have gained immense importance in the last two decades. This development was supported by the invention of mobile communications, but it was especially boosted by the enormous expansion of the Internet. Applications like video-streaming, file sharing, telecommuting or video-conferencing awakened the desire for increasing data rates, resulting in a hundred-fold increase from some hundred kbit/s to around several ten Mbit/s in the last twenty years. Today, the necessity of sufficiently high data rates is as present as ever. With High Definition Television (HDTV) gaining momentum and the tendency to use cloud services and connect everything in the Internet of Things customers still long for reliable and suitable connections.

Basically, there are three different options to bring broadband services to the customers. Fully fiber-based networks would be the technically most advanced solution providing the highest available data rates. But optical equipment and trenching of cables is extremely costly, making it economically non-profitable for telecommunication companies.

As a second option, Long Term Evolution (LTE) can be considered as an alternative to fiber for provision of broadband services. Thanks to achievable data rates and comparably low cost of installation, this is definitely true for undeveloped and sparsely populated areas. In urban areas however, where lots of customers share the available bandwidth, mobile services can easily come at the expense of reliability and connection quality.

The third very valuable option is Digital Subscriber Line (DSL) [GDJ06, SSCS03, SCS99]. Digital subscriber line technology offers wireline broadband services by enabling digital communications over the existing telephone infrastructure. The telephone lines were originally designed for voice communications at frequencies of up to some kHz providing limited data rate. DSL simply widens the occupied transmission bandwidth, resulting in an enormous gain in performance. In addition, DSL has the great advantage of profiting from the copper telephone network, which is widely expanded and contains several hundred million loop plants. By occupying that existing infrastructure it has lower implementation cost than fiber and due to the fact that each user occupies a single copper line it can provide highly reliable services.

Since the development in the late 1980's, DSL has seen many improvements. In the first fifteen years High Data Rate Digital Subscriber Line (HDSL), Symmetric Digital Subscriber Line (SDSL) and Asymmetric Digital Subscriber Line (ADSL) were designed and operated in the market. Starting with HDSL and its extension SDSL, symmetric services with data rates of some Mbit/s were provided to the customers first on two twisted-pairs and then also on one twisted-pair. In the late 90's, the more consumer-oriented ADSL was introduced triggered by the Internet boom. With higher Downstream (DS) than Upstream (US) rates, it perfectly meets the Internet users' requirement for asymmetrical service as they normally download more than they upload. ADSL is able to provide DS rates of nearly 10 Mbit/s and a tenth of that in US. An evolutionary step followed in the mid 2000's with the invention of Very High Speed Digital Subscriber Line (VDSL). By then, fibers attached the network core to the Central Office (CO) and DSL services connected the CO to the customer premises, which resulted in loop lengths of some kilometers. With VDSL, the fiber was coming closer to the customer. Fibers were laid to the end of the street where an Optical Network Unit (ONU) was installed. As a result, VDSL only runs on the twisted-pair connection between the ONU and the customer premises and loop lengths were reduced. The shortened cable lengths allowed occupancy of much higher frequencies and enabled an increase in data rate to some tens of Mbit/s. The successor of VDSL, Very High Speed Digital Subscriber Line 2 (VDSL2), still operates on the last mile between customer premises and fiber-network core, but at an even higher bandwidth.

Modern DSL communication systems like VDSL and VDSL2 use the frequency band up to several ten MHz on each copper cable to offer much higher data rates than achievable with the voice bands. Unfortunately, the twisted pairs of the different users are bundled within large cable binders typically containing 20 to 100 individual pairs. Due to the high frequencies and non-perfect insulation of the twisted pairs there is significant electromagnetic coupling among nearby lines. This leads to severe interferences between the different users transmitting within a binder, resulting in two different kinds of Crosstalk (XT): Near-End Crosstalk (NEXT) and Far-End Crosstalk (FEXT). Near-end crosstalk occurs when transmit signals of one stream direction disturb the received signals of the other stream direction. It can be easily avoided by the use of duplexing methods. Far-end crosstalk results when the transmit signals of the same stream direction interfere. It can be up to 20 dB larger than the background noise and is the major performance limiter between adjacent lines in a cable binder. Because of that it is very reasonable and highly effective to mitigate the far-end

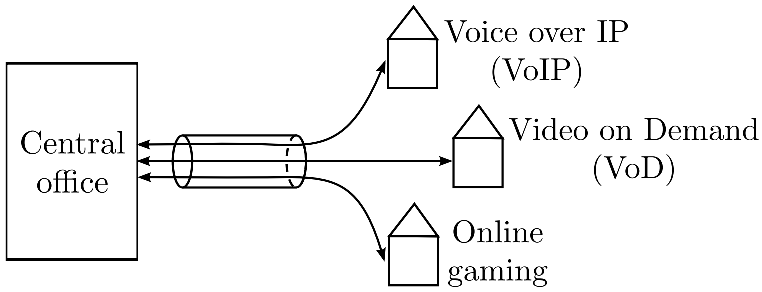


Figure 1.1: Multimedia applications in DSL access networks

crosstalk interferences.

Principally, the negative impact of FEXT can be minimized in two ways. Far-end crosstalk can either be eliminated by crosstalk cancellation or reduced by spectrum management [Rua14, Cen04, Yu02]. Digital subscriber line systems are multicarrier systems, where the bandwidth is split into a large number of narrowband transmission channels. They apply Discrete Multi-Tone Transmission (DMT), a technique which is identical to Orthogonal Frequency Division Multiplexing (OFDM) [Roh11]. Spectrum management exploits this fact by reasonably shaping the transmit spectra on the different lines. By adapting the transmit power on the subchannels of all lines in the binder, less mutual interferences are produced. The crosstalk influence is decreased but can never be completely avoided. In addition, spectrum management is not advantageous in every transmission scenario.

Crosstalk cancellation tackles the problem with FEXT by signal processing. For that purpose, DSL systems are modeled as multi-user systems where every user in the binder is considered as a transmitter and also as a recipient of the signals of all other lines. When full Channel State Information (CSI) of both direct and crosstalk channels is known with adequate precision and accuracy, crosstalk cancellation and precoding techniques are able to completely remove the interferences from the other lines. Especially methods based on Zero-Forcing (ZF) are known for eliminating FEXT in a near-optimum way. However, crosstalk elimination procedures might also need an unbearable amount of computational complexity to achieve crosstalk-free transmission for large binder sizes. Generally, only limited computational resources are available and FEXT can only be partially removed.

Nowadays, DSL systems have to be able to support an always increasing amount of high-rate applications. At the same time, they face services with highly differing data rates and as a result users with highly varying data rate demands (see Fig. 1.1). If DSL users want to stably run several high-rate applications in parallel, they will need guarantees on their achievable data rates. Partial crosstalk cancellation procedures could be a solution, but existing algorithms select the canceled crosstalkers by focusing on capacity maximization and do not take into account the data rate demands the customers really have. This makes service providers either not being able to give data rate guarantees to their customers or spending more computational complexity than actually needed for target achievement on a binder.

In the following two important problems of DSL systems are addressed: Firstly, the issue of estimating the crosstalk channels for crosstalk cancellation and precoding is discussed as system performance strongly depends on the quality of channel estimation. Both US and DS channels need to be initially estimated with sufficiently high accuracy. To always maintain the full performance of the DSL system the precoder and canceler coefficients need an update over time as the DSL channel can vary slowly due to temperature and humidity changes. Channel estimation and update procedures are introduced which combine the good performance of pilot-based estimation techniques with low pilot and signaling overhead.

Secondly, another goal is to find crosstalk mitigation methods which satisfy the need of setting and supporting data rate requirements in DSL access networks at a limited amount of computational complexity available for a binder. In a first step, novel successive selection algorithms for partial crosstalk cancellation and precoding are found, which are able to adapt the users' data rates to the desired values. In a second step, the successive selection methods are combined with spectrum management techniques to support high data rate targets with a low amount of computational complexity more efficiently.

The structure of the thesis is given as follows: In Chapter 2, DSL systems are considered in a single-user environment. The chapter gives an overview of techniques and methods applied on a single twisted-pair and describes the transmission channel of a single line.

Chapter 3 extends the description of DSL systems to the multi-user environment, where crosstalk is not modeled as noise, but as an interference channel. Properties of multi-user up- and downstream transmission at the presence of crosstalk are explained and the considered crosstalk channel model is presented.

The crosstalk cancellation techniques which are the basis for all proposed methods, are introduced in Chapter 4. Upstream full crosstalk cancellation and downstream full precoding completely eliminate crosstalk. Partial crosstalk cancellation and partial crosstalk precoding do not remove the influence of crosstalk in total, but just reduce it. Only the interferences of a set of selected crosstalkers are eliminated.

Spectrum management is explained in Chapter 5. For proper understanding, the spectrum management problem and existing solutions to it are presented.

The contributions and results of this thesis are given in Chapter 6 and especially in Chapter 7. Chapter 6 explains the proposals for crosstalk channel estimation and crosstalk channel update and gives performance results. Chapter 7 addresses the problem of data rate constraints in DSL access networks and limited computational resources for crosstalk cancellation procedures. Two methods are presented, which allow users in a cable binder to obtain their data rate targets with a limited computational complexity. The first method uses partial crosstalk cancellation and precoding and selects the set of eliminated crosstalkers based on their data rate targets. The second proposal extends the first method by combining it with spectrum management. For both procedures performance results are given. All contributions and their corresponding results are summarized in the conclusion in Chapter 8.

II

Digital Subscriber Line Transmission

2.1 Basics

The DSL technology allows the transport of high-bit rate digital information over conventional old copper telephone lines. Essentially, a digital subscriber line is the analog twisted-wire pair connection between the CO and the customer premises (see Fig. 2.1), also referred to as the local loop.

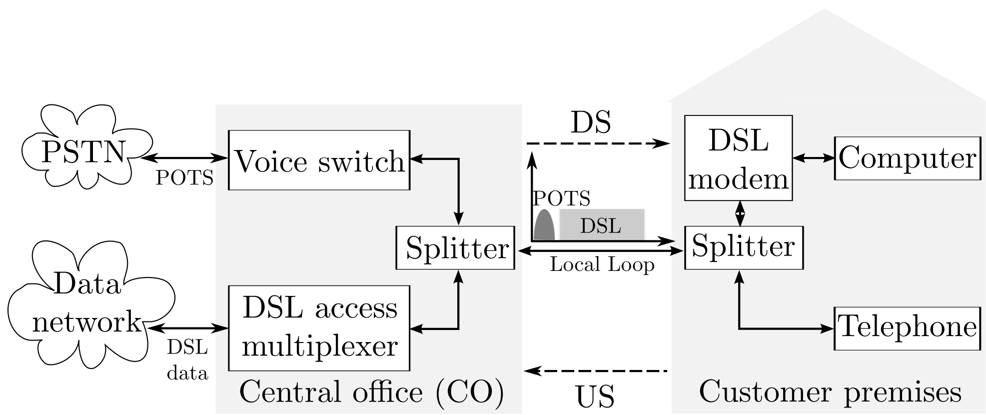


Figure 2.1: DSL reference model for a single-line situation

A DSL transmission can be either established in the US direction, i. e. from the customer premises to the CO, or in the reverse DS direction, since all DSL services provide both streaming ways. As the initial use of the twisted pairs was the analog telephony, the local loop carries both the signal of the Plain Old Telephone Service (POTS) and the DSL data signal, where POTS occupies the baseband from 200 Hz to 4 kHz and DSL uses frequencies up to 30 MHz in the copper line. To separate the telephone signal from the DSL service, DSL filters (splitters) are needed on both sides of the loop. At the customer premises, the voice signal is directed to the telephone after the splitter. At the CO, a voice switch directs the voice signal into the Public Switched Telephone Network (PSTN). To process the DSL signal, modems are required on both ends of the copper line. On the side of the customer

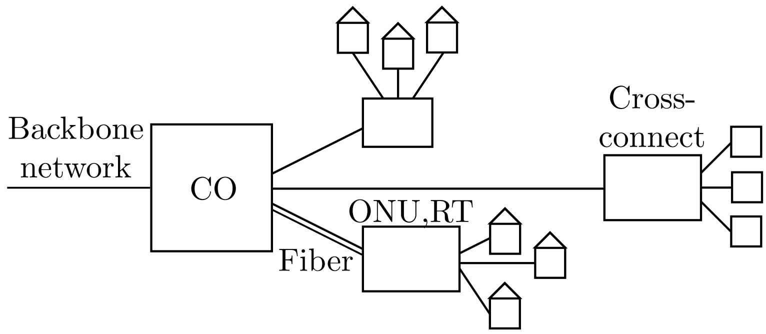


Figure 2.2: Typical deployment scenario

premises, the DSL modem connects the loop to a computer, by which the data is generated for upstream transmission or accessed in downstream. In the CO at the other end of the loop, a Digital Subscriber Line Access Multiplexer (DSLAM) provides DSL service to the users' premises. It usually contains many line cards serving multiple customers and is the aggregation point of many DSL lines. The DSLAM routes the DSL signals from individual subscribers' phone lines and multiplexes the data to transmit it in the backbone network.

A schematic picture of the telephone access network is shown in Fig. 2.2. Individual wire pairs are generally grouped together in binder-groups of 4 or 10 cables. Fifty to 100 wire pairs are bundled together into a cable. Most telephone lines start at the CO, where over $100,000$ customers can be served, if the CO is large. Usually, larger feeder cables with several thousand wire-pairs emanate from it. They merge in street cabinets, which are cross-connections between large feeder cables and smaller distribution cables. In addition, there can be fiber-to-the-cabinet solutions, where the optical fibers are drawn out to the cross-connection points, which are then called optical network units or ONUs.

2.2 *Discrete Multi-Tone Transmission Technique*

The transmission channel of a copper twisted-pair strongly depends on frequency, as exemplarily shown in Fig. 2.3. With increasing frequency there is a heavy growth in line attenuation and variation of delay. Together they cause Intersymbol Interferences (ISI) as successively transmitted symbols interfere with each other. Especially for broadband systems such as DSL, hundreds of symbols can be affected. This leads to a vast degradation of performance when no equalization is applied.

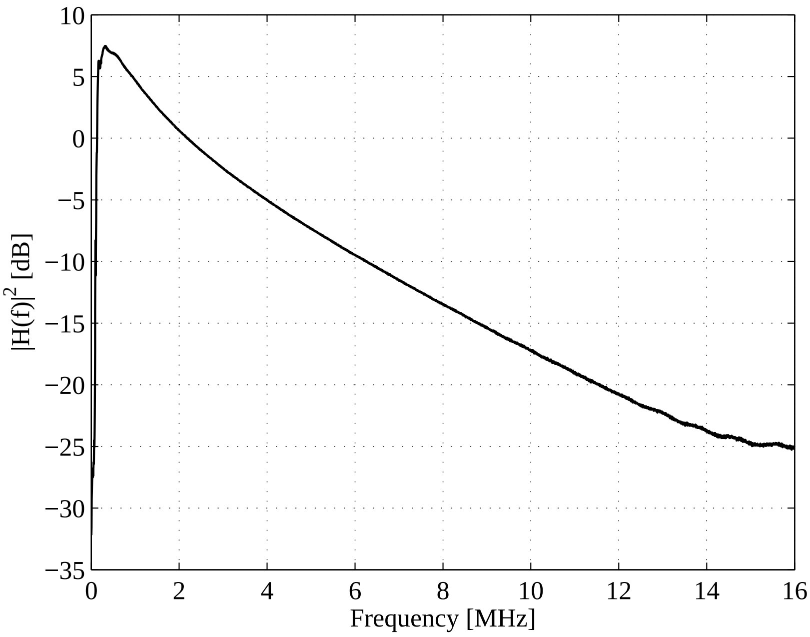


Figure 2.3: Measured direct channel transfer functions

Single carrier DSL systems like HDSL [WS91] or SDSL [Bha99] can for instance use a Decision Feedback Equalizer (DFE) [BP79] at the receiver side to remove the ISI. However, the increase in quality comes at the cost of higher run-time complexity and the possibility of error propagation. An alternative is a Tomlinson-Harashima Precoder (THP) [Tom71, HM72] at the transmitter to precompensate for ISI, which does not suffer from error propagation. Unfortunately, it requires accurate channel knowledge at the transmitter, which leads to a high transmission overhead due to signaling and increased computational complexity because of the channel measurements.

Modern DSL systems such as ADSL and VDSL can also be discrete multi-tone modulated. Discrete multi-tone transmission [PR80, RCK92, Roh11] is a multicarrier technique, which was invented to address the shortcomings of the single-carrier systems in terms of the frequency-selective channel. By splitting the broadband transmission channel into several narrowband subchannels, it is able to avoid ISI with much lower complexity than the single-carrier systems, with neither error propagation nor increased signaling overhead. Figure 2.4 illustrates the principle of DMT for a frequency-selective channel.

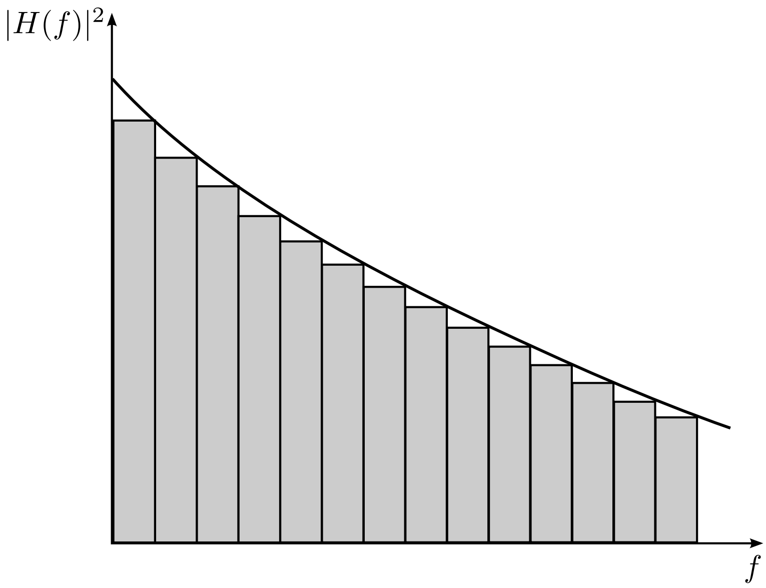


Figure 2.4: Principle of DMT

2.2.1 Discrete Multi-Tone Modulation

Discrete multi-tone modulation is a multicarrier scheme and the baseband alternative to OFDM in radio systems (like e. g. in Digital Audio Broadcasting (DAB) [FJK+02], Digital Video Broadcasting (DVB) [Rei04]). The transmission channel is divided into many parallel subchannels, also named tones. Within a subchannel, the channel response can be assumed being flat so that ISI do not occur. The modulation of the transmit signal is done by an Inverse Discrete Fourier Transform (IDFT), which was originally described in [WE71]. A single DMT symbol is given by

$$x_i = \frac{1}{\sqrt{\bar{K}}} \sum_{k=1}^{\bar{K}} X_k \cdot e^{j(2\pi/\bar{K})(k-1)(i-1)}, \ \forall i \in [1, \bar{K}], \tag{2-1}$$

where $\bar{K}$ is the size of the IDFT, i is the index of the time-domain samples, k is the tone index and X_k are the complex modulated frequency-domain inputs.

Since DSL can be considered a baseband transmission, the generated signal is real-valued and can be applied directly to the channel after the digital-to-analog conversion. The input sequence X_k, $k = 1, \ldots, \bar{K}$ in Eq. 2-1 is Hermitian symmetric to generate the real-valued output. Only half of the IDFT components are available to transport useful data. Consequently, the number of useful tones K is given by $K = \bar{K}/2$ [Cio91].

In practice, the frequency response across the selected tones is not perfectly flat. This results in imperfect elimination of the ISI by the partitioning process. To completely prevent the receive signal from intersymbol interferences and also avoid interchannel interferences, a cyclic prefix is appended at the beginning of every DMT symbol before transmission. It is a copy of the last ν time-domain samples, increasing the number of samples in the transmit symbol to $\bar{K}+\nu$. For the purpose of maintaining orthogonality and avoiding ISI, the cyclic prefix has to have at least the length of the maximum channel tap delay. It ensures that the DMT system is still equal to $\bar{K}$ parallel tones after transmission over the DSL channel. This simplifies the equalization to only a one-tap frequency equalizer for each DMT tone. A scalar multiplication is sufficient to equalize each subchannel. On the receiver side, the frequency spectrum is calculated using the Discrete Fourier Transform (DFT) by

$$Y_k = \frac{1}{\sqrt{\bar{K}}} \sum_{i=1}^{\bar{K}} y_i \cdot e^{-j(2\pi/\bar{K})(k-1)(i-1)}, \; \forall k \in [1, \bar{K}]. \tag{2-2}$$

In frequency domain, the transmission on each tone can be mathematically described by

$$Y_k = H_k \cdot X_k + N_k, \tag{2-3}$$

with transmit symbol X_k, received symbol Y_k, channel coefficient H_k and noise N_k. The noise component includes thermal noise, Radio Frequency Ingress (RFI), and alien noise. Consequently, all useful tones can be equalized in frequency domain with the help of a Frequency Domain Equalizer (FEQ) according to

$$\hat{X}_k = \frac{Y_k}{\hat{H}_k}, \tag{2-4}$$

with the estimated channel coefficient $\hat{H}_k$.

A simplified block diagram of a DMT system is given in Fig. 2.5. On the transmitter side, K modulation symbols are generated out of the bit stream. On each independent tone a Quadrature Amplitude Modulation (QAM) is applied. In DSL systems, the efficient computational methods of the IDFT and DFT, the Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT), are used to keep the computational complexity low. The time-domain transmit signal is generated by the IFFT. An $\bar{K}$-size IFFT calculates the consecutive time-domain samples. The cyclic prefix is attached before conversion to the analog domain.

On the receiver side, the cyclic prefix is removed from the received signal. The frequency-domain symbols are calculated efficiently with the FFT and the transmit symbols are reproduced by frequency-domain equalization. The demapper recovers the data stream.

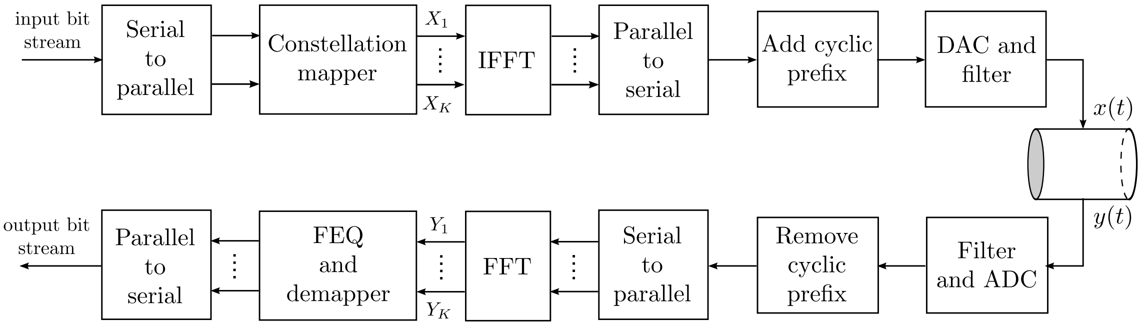


Figure 2.5: DMT transmitter and receiver block diagram

2.2.2 Data Rate Calculation

In a multicarrier system, the data transmitted on each subchannel contributes to the overall achievable data rate: the total data rate is the sum of the data rates on the subchannels.

The theoretical limit for the transmission rate over a given channel is called the capacity [Pro01]. On the kth tone of a DMT system and under presence of Additive White Gaussian Noise (AWGN) the capacity C_k is

$$C_k = \Delta f \cdot \log_2(1 + SNR_k) \tag{2-5}$$

where Δf is the subchannel spacing and SNR_k is the Signal-to-Noise Ratio (SNR) on subchannel k.

As the channel capacity cannot be reached by real DSL systems, the Shannon gap needs to be considered for the calculation of the DMT system data rate. It is a measure of the degradation in performance of real systems relative to the theoretically optimum Shannon capacity. Additionally, DSL systems are conventionally operated with some margin to guarantee good performance of the system, which increases the gap. Channel coding can be applied, which improves performance and reduces the gap again. Considering all this, the SNR gap to capacity is defined by [GDJ06]

$$\Gamma = \frac{\Gamma(P_e) \cdot \gamma_m}{\gamma_c} \tag{2-6}$$

with γ_m being the noise margin and γ_c being the coding gain. $\Gamma(P_e)$ describes the Shannon gap at a given bit error rate P_e.

Together with the SNR gap, the number of bits that can in practice be transmitted on a tone per symbol is

$$b_k = \log_2\left(1 + \frac{SNR_k}{\Gamma}\right). \tag{2-7}$$

The overall number of bits b that can be transmitted in the complete DMT system then sums up to

$$b = \sum_{k=1}^{K} \log_2 \left(1 + \frac{SNR_k}{\Gamma}\right) \tag{2-8}$$

per symbol duration.

To obtain the data rate out of the number of bits per symbol, the DMT symbol rate needs to be considered. DMT systems require application of a cyclic prefix to avoid ISI as described in Sec. 2.2.1. This results in a DMT symbol rate slightly lower than Δf. The total achievable data rate of the multicarrier system is then given by

$$\begin{aligned} R =& f_S \cdot b \\ =& f_S \cdot \sum_{k=1}^{K} \log_2 \left(1 + \frac{SNR_k}{\Gamma}\right) \end{aligned} \tag{2-9}$$

where f_S is the DMT symbol rate.

2.2.3 Bit and Power Loading

The ability to adapt to different channel conditions by bit and power loading strongly contributed to the grand success of multicarrier DSL systems. Figure 2.3 showed, that the channel quality on the multiple tones in a DMT system heavily varies over frequency. As DSL channels do not change quickly, large performance gains are enabled, when adaptive modulation is applied globally on the subchannels. The size of the constellation on each tone is adjusted according to the SNR experienced on that subchannel, which allows full exploitation of the high SNR that twisted-pairs can offer and application of constellations based on 10 to 12 bits.

In multicarrier transmission systems, information and power is assigned to the subchannels by loading algorithms. The bit loading on a tone is the number of bits that is transmitted per symbol on each DMT-subchannel. The receiver measures the SNR on each tone and reports it back to the transmitter, which can then adaptively vary the constellation used for transmission. It allows DMT systems to achieve high spectral efficiency by optimizing the distribution of bits over available bandwidth. In addition, modems are able to vary the power adapted to each tone. The process of assigning power to the subchannels based on a certain objective is called power loading.

There are two loading problems of interest: maximization of the data rate for a fixed total power and minimization of the power for a fixed given data rate. Both can be solved by the waterfilling solution [Ran08, TV05]. It is the basis for the algorithm presented in Sec. 7.2, which was developed in this thesis.

2.2.3.1 Waterfilling solution

The first optimization problem of maximizing the data rate R requires maximization of the number of loaded bits b depending on the number of loaded bits b_k per tone and the corresponding power P_k per tone. Based on the concept of capacity and the calculation of the data rate given in Eq. 2-9, the maximum number of bits that can be transmitted over the set of tones should satisfy the following condition:

$$\max b(P_k, b_k) = \max \left\{ \sum_{k=1}^{K} b_k \right\} \tag{2-10}$$

$$= \max \left\{ \sum_{k=1}^{K} \log_2 \left(1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \sigma^2} \right) \right\} \tag{2-11}$$

$$\text{subject to: } \sum_{k=1}^{K} P_k = P_{max} \tag{2-12}$$

The maximum overall transmit power P_{max} on a line is constrained by the analog hardware of the DSL modems.

To solve the optimization by applying Lagrangian multipliers, the cost function which combines the optimization criterion in Eq. 2-11 with the constraint in Eq. 2-12 is

$$\mathcal{L}(\lambda, P_1, \ldots, P_k) = \sum_{k=1}^{K} b_k - \lambda \sum_{k=1}^{K} P_k \tag{2-13}$$

$$= \sum_{k=1}^{K} \log_2 \left(1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \cdot \sigma^2} \right) - \lambda \sum_{k=1}^{K} P_k. \tag{2-14}$$

Differentiating Eq. 2-14 over P_k and using the constraint to get the optimum transmit Power Spectral Density (PSD), leads to the so called Rate-Adaptive (RA) waterfilling solution. The power which should be loaded on the tones to achieve the maximum data rate is given by

$$P_k = \frac{1}{K} \left(P_{max} + \sum_{k=1}^{K} \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \right) - \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \tag{2-15}$$

$$= \mu_{ra} - \frac{\sigma^2 \cdot \Gamma}{|H_k|^2}. \tag{2-16}$$

The loaded power becomes the sum of a constant μ_{ra} and a frequency-dependent power value. μ is called the waterfilling level, which defines the optimal point of power transmitted on all tones k. It is constant over frequency. μ_{ra} is the waterfilling level for rate-adaptive waterfilling.

The second loading problem of interest is the minimization of the needed total power P to achieve an overall constraint in loaded bits $b^{(T)}$. The minimum power that can be allocated to the tones should satisfy the following condition

$$\min P(b_k, P_k) = \min \left\{ \sum_{k=1}^{K} P_k \right\} \tag{2-17}$$

$$\text{subject to: } \sum_{k=1}^{K} \log_2 \left(1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \cdot \sigma^2} \right) = b^{(T)}. \tag{2-18}$$

The waterfilling solution for Power-Adaptive (PA) waterfilling results in

$$P_k = 2^{\frac{1}{K}\left[b^{(T)} - \sum\limits_{k=1}^{K} \log_2 \left(\frac{|H_k|^2}{\Gamma \cdot \sigma^2} \right) \right]} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \tag{2-19}$$

$$= \mu_{pa} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \tag{2-20}$$

with waterfilling constant μ_{pa}. Equations 2-16 and 2-20 show that the solutions to the different loading problems vary only in the waterfilling levels. The complete derivations for both loading problems can be found in Appendix A.1. Algorithms for simulation can be found in Appendix A.2. Figure 2.6 illustrates the principle of waterfilling.

In practice, the transmission power can only have positive values. Therefore, when the waterfilling level μ is smaller than $(\sigma^2 \cdot \Gamma)/|H_k|^2$ like for the first tone in Fig. 2.6, the transmission power on that tone is clipped to zero, the number of used tones is reduced to K^* and the water level for all other tones is raised. This is the case when the corresponding tone has such bad channel quality that data should not be transmitted on it.

The allocated power on each tone is then defined by

$$P_k^+ = \begin{cases} \mu - \frac{\sigma^2 \cdot \Gamma}{|H_k|^2}, & \text{if } \frac{\sigma^2 \cdot \Gamma}{|H_k|^2} < \mu \\ 0, & \text{if } \frac{\sigma^2 \cdot \Gamma}{|H_k|^2} \geq \mu \end{cases} \tag{2-21}$$

and the number of transmitted bits is calculated as

$$b_k^+ = \begin{cases} \log_2 \left(1 + \frac{1 + SNR_k}{\Gamma} \right), & \text{if } \frac{\sigma^2 \cdot \Gamma}{|H_k|^2} < \mu \\ 0, & \text{if } \frac{\sigma^2 \cdot \Gamma}{|H_k|^2} \geq \mu. \end{cases} \tag{2-22}$$

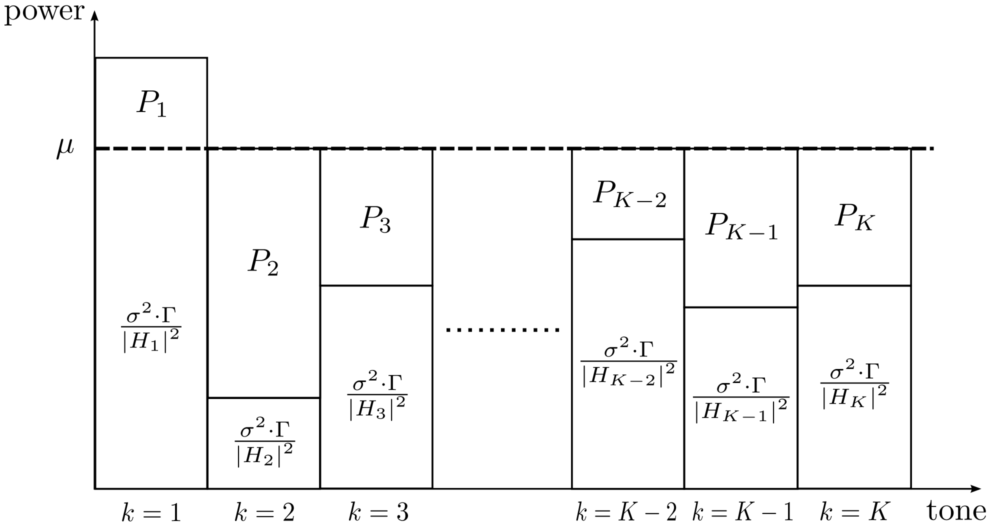


Figure 2.6: Waterfilling principle

For DSL systems, there may also exist spectral masks to guarantee spectral compatibility with other systems transmitting on the telephone line [Cen04]. When a DSL system has to use a spectral mask, the transmission power is limited and results in

$$P_k^{+,mask} = \begin{cases} \left[\mu - \frac{\sigma^2\cdot\Gamma}{|H_k|^2}\right]_0^{P_k^{mask}}, & \text{if } \frac{\sigma^2\cdot\Gamma}{|H_k|^2} < \mu \\ 0, & \text{if } \frac{\sigma^2\cdot\Gamma}{|H_k|^2} \geq \mu \end{cases} \tag{2-23}$$

The calculated transmit power allocation can be directly applied in reality. But the fractional values for b_k, as calculated by the waterfilling solution cannot be implemented in real applications. Proposals presenting algorithms for practical implementation which solve the loading problem with integer numbers of bits are given in [CCB95, FH96, Cam98, Cam99, GA04, Lev01, RF09].

2.3 Duplexing

Almost all DSL services require bidirectional transmission of data. To send information US and DS on a single line, either a duplexing method or Echo Cancellation Hybrid (ECH) is needed to separate transmitted and received signals.

In bidirectional transmission of data, reflections of the transmitted signal can produce noise or echoes into the near-end receiver, which can be up to 20 dB larger than the desired received

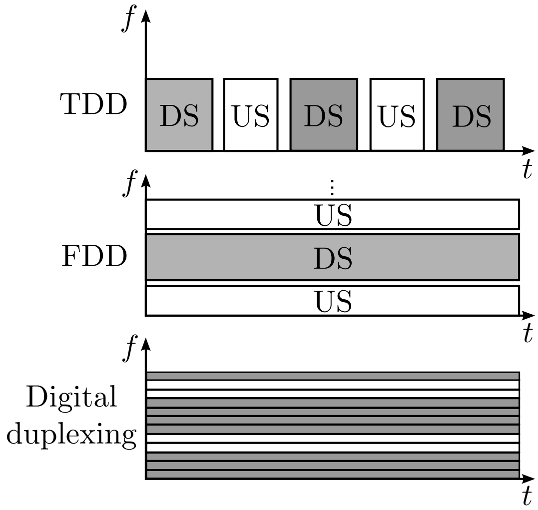


Figure 2.7: Duplexing methods

signals. Echo cancellation is thus necessary to allow bidirectional transmission of data. An echo canceler rejects the echo by adaptively generating replicas of it and subtracting them from the received signal [SCS99].

Most DSL systems use duplexing methods to separate the signals in opposite stream directions. The traditional duplex methods are Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD), followed by Digital duplexing which came up with VDSL [GDJ06, GDJ08, Sjö00].

When TDD is applied, the wireline channel is either used for US or DS communication. Each direction are allocated disjoint time durations in which they can use the channel. Only transmitter or receiver are switched on at a time, which saves power and avoids the need for an echo canceler. But data rate is lost as only parts of the connection time is available for transmission. Additionally, the turnaround of the loop to the other stream direction requires a silent guard time in practical systems, in which no data can be transmitted. Time division duplexing can be used with single-carrier and multi-carrier modulation. All lines in the cable binder must use the same US/DS allocation.

Frequency division duplexing transmits different stream directions in non-overlapping frequency bands. It is the most widespread method of duplexing used in DSL. Same as TDD, FDD can be combined with any type of modulation method. For instance, ADSL uses FDD in combination with DMT. Frequency division duplexing does not rely on system-wide synchronization of the modems like TDD, but needs analog filters in the transmit and receive path to ensure that only minimal amounts of the transmit signal leak into the receiver. With

increasing number of frequency bands, this results in an increase in complexity. In addition, FDD needs guard bands which prevent effective usage of the bandwidth. A method which overcomes the loss of data rate introduced by the guard bands is digital duplexing also known as Zipper. Digital duplexing can be thought of as an extension of FDD with a very large number of frequency bands. It is specific to DMT modulation as it divides the bandwidth by assigning different tones to US and DS. As generally known from DMT systems, the bands are overlapping, but mathematically orthogonal to each other. Therefore interferences are not caused outside the bands. Similarly to TDD, all modems must be synchronized, so that transmission of new DMT frames starts simultaneously in each modem. In addition to the cyclic prefix, a cyclic suffix is needed to allow alignment of the symbol boundaries of both stream directions.
Fig. 2.7 compares TDD, FDD and digital duplexing in a time-frequency diagram. In this thesis, only FDD-based systems are considered.

2.4 The Wireline/Twisted-Pair Channel

The channel transfer function $H(f)$ describes the effects on a signal when it is sent via a transmission channel. In order to derive the channel transfer function of a DSL twisted-pair, the characteristics of the twisted-pair telephone line have to be studied. The twisted-pair line is modeled as a transmission line, as many characteristics of the twisted-pair wire can be derived from traditional transmission line theory.
Fig. 2.8 shows a model for an infinitesimally small part of a transmission line with which the direct channel of a DSL line can be accurately estimated. The behavior of the line can be characterized by the primary parameters: resistance R $[\Omega/km]$, inductance L $[H/km]$, capacitance C $[F/km]$, and conductance G $[Mho/km]$. All parameters are assumed to be independent of length or position and usually vary with frequency. The primary parameters

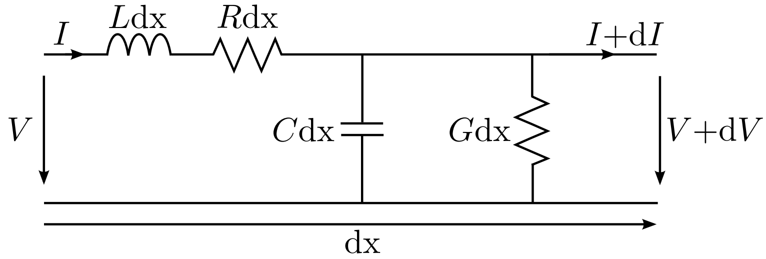


Figure 2.8: Line section of length dx

Table 2.1: Secondary cable parameters

	TP1 (⌀0.4 mm)	**TP2 (⌀0.5 mm)**
Resistance		
r_{0c}	286.17578 Ω/km	174.55888 Ω/km
a_c	0.1476962	0.053073481
Inductance		
l_0	675.36888 μH/km	617.29539 μH/km
l_∞	488.95186 μH/km	478.97099 μH/km
b	0.92930728	1.1529766
f_m	806.33863 kHz	553.760 kHz
Capacitance		
c_∞	49 nF/km	50 nF/km
Conductance		
g_0	43 nS/km	234.87476 fS/km
g_e	0.70	1.38

are defined as

$$R(f) = \sqrt[4]{r_{0c}^4 + a_c \cdot f^2} \tag{2-24}$$

$$L(f) = \frac{l_0 + l_\infty \left(\frac{f}{f_m}\right)^b}{1 + \left(\frac{f}{f_m}\right)^b} \tag{2-25}$$

$$C(f) = c_\infty \tag{2-26}$$

$$G(f) = g_0 \cdot f^{g_e}. \tag{2-27}$$

The secondary parameters r_{0c}, a_c, l_0, l_∞, b, f_m, c_∞, g_0 and g_e depend on cable diameter, material and construction. Table 2.1 shows the secondary parameters for standard twisted-pair wire types TP1 and TP2.

The transmission on a DSL line can be modeled by a two-port linear circuit (see Fig. 2.9). The ABCD matrix of a loop is known to be [Roh11]

$$\mathbf{A} = \begin{bmatrix} cosh(\gamma d) & Z_0 \cdot sinh(\gamma d) \\ \frac{1}{Z_0} \cdot sinh(\gamma d) & cosh(\gamma d) \end{bmatrix} \tag{2-28}$$

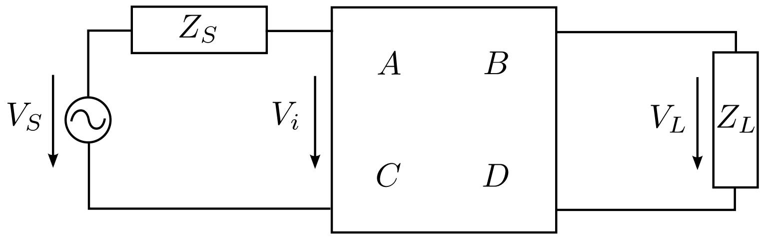


Figure 2.9: 2-port model of DSL line transmission

with line length d. The characteristic impedance Z_0 of the transmission line is defined as

$$Z_0 = \sqrt{\frac{R + j2\pi f L}{G + j2\pi f C}}. \tag{2-29}$$

and the propagation constant γ can be calculated by

$$\gamma(f) = \sqrt{(R + j2\pi f L)\,(G + j2\pi f C)}. \tag{2-30}$$

The transmission line is connected to a source V_S with source impedance Z_S and terminated with load Z_L. The channel transfer function $H(f,d)$ for a twisted-pair cable (also called the insertion gain transfer function) is the voltage V_L across the load Z_L with the transmission line inserted, divided by the voltage V_{no} across load Z_L with no transmission line. Together with the ABCD parameters, the channel transfer function is given by

$$H(f,d) = \frac{V_L}{V_{no}} = \frac{Z_L + Z_S}{Z_L \cdot \cosh(\gamma d) + Z_0 \cdot \sinh(\gamma d) + \frac{Z_S \cdot Z_L}{Z_0} \cdot \sinh(\gamma d) + Z_S \cdot \cosh(\gamma d)}. \tag{2-31}$$

The channel transfer function is frequency-dependent and furthermore changes with cable length d.

Matching the source and the load impedance to the characteristic impedance of the transmission line is a reasonable assumption to minimize the reflection coefficient. When the line is ideally terminated with Z_0, so that $Z_L = Z_0 = Z_S$, the channel transfer function simplifies to [Wer91]

$$H(f,d) = e^{-\gamma(f) \cdot d}. \tag{2-32}$$

2.5 Overview of xDSL Standards

To keep up with competitive services such as mobile providers and to accommodate for the steadily growing demands for data rates, DSL services develop at a fast pace. The range of

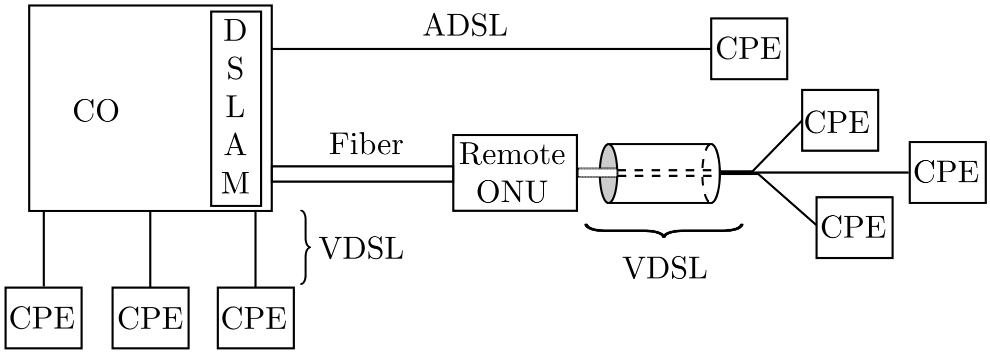


Figure 2.10: Typical VDSL deployment architecture

DSL technologies is quite broad, therefore the main purpose of this section is to give a brief overview of the most popular DSL standards [GDJ06, SSCS03].

Integrated Services Digital Network (ISDN) can be regarded as the first DSL technology. The focus was on transmission of voice signals and low-speed data signals. Like POTS, it is a lifeline service, where the phone is powered via the line and function is even guaranteed in the event of a power failure.

HDSL is used for wideband digital transmission within corporate sites and as private lines from CO to customer premises. Generally, two lines are used for transmission. It provides symmetrical service and due to its bandwidth allocation from 0 Hz on, it does not allow POTS at baseband. SDSL is the extension of HDSL to just one line.

ADSL simultaneously transports asymmetric data services and POTS on the line by using the frequency spectrum above the voiceband for data transmission. ADSL2 uses the frequency band up to 2 MHz. It was especially designed to exploit the one-way nature of most multimedia communications, where large amounts of information is received in the downstream direction and only small amounts of control data are returned upstream. It is the first DSL system applying multicarrier modulation. Due to line length restrictions of about 3.7 km, it is generally found in urban areas.

VDSL is an extension of ADSL in terms of symmetric services and higher bit rate. However, provision of higher bit rates is only possible over shorter loop lengths as VDSL uses the frequency spectrum up to 12 MHz and VDSL2 even uses the frequency band up to 30 MHz. Figure 2.10 shows a typical deployment architecture of VDSL compared to ADSL. Customer premises receiving VDSL (or higher) services are either directly connected to the CO or are connected via a remote optical network unit, which is connected via fiber to the CO. Table 2.2 briefly summarizes the most important characteristics of different DSL services.

Table 2.2: Overview DSL standards

Standard	Ratified	Data Rate US	Data Rate DS	Modulation	Number of Pairs	Duplex	Loop reach
ISDN	1985	Up to 160 kbit/s	Up to 160 kbit/s	Single carrier	1	ECH	Up to 5.5 km
HDSL	1992	2.048 Mbit/s (one-way) 1.544 Mbit/s	1.544 Mbit/s	Single carrier DMT, CAP	mostly 2	ECH	Up to 5 km
SDSL	1995	Up to 2.3 Mbit/s	Up to 2.3 Mbit/s	Single carrier	1	FDD	Up to 3.5 km
ADSL	1999	Up to 1 Mbit/s	Up to 9 Mbit/s	DMT	1	FDD, ECH, TDD	Up to 3.7 km
VDSL	2004	Symmetric service: 26 Mbit/s Asymmetric service: 6.4 Mbit/s	 26 Mbit/s 52 Mbit/s	DMT, CAP	1	FDD, ECH	Up to 1.5 km
VDSL2	2005	Without crosstalk reduction: 10 Mbit/s With crosstalk reduction: 40 Mbit/s	 50 Mbit/s 100 Mbit/s	DMT	1	FDD	Up to 1.5 km

III

Multiuser Transmission

DSL transmission is typically established between a central node and the Customer Premises Equipment (CPE) of a remote user. The central node may be a CO, an ONU or Remote Terminal (RT), depending on the application. However, in this thesis a DSL topology with a CO as central node is considered (see Fig. 3.1).

The CO and the locally distributed CPEs are connected via twisted pair lines, each line belonging to one user. When information is transmitted from the CO to the different users, a DS communication was set up. Transmission of information in the opposite direction is referred to as US communication.

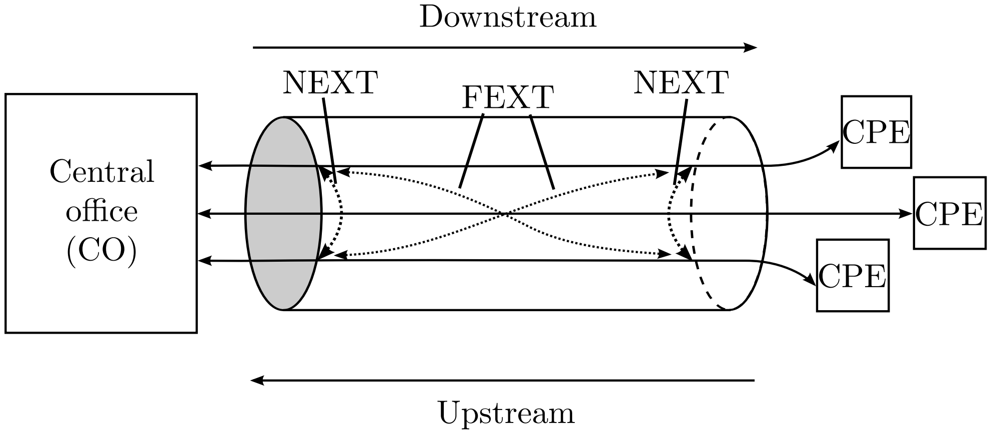


Figure 3.1: Multi-user transmission

Initially, the DSL environment was thought of as a single-user environment, in which each user occupies his own twisted copper pair and modems are synchronized and transmit simultaneously. As a fact, the DSL environment is a multi-user environment as for a large part of their length the twisted pairs are physically close to each other because they are bundled in a cable binder. Non-perfect insulation of the lines leads to electromagnetic coupling between them and results in mutual interferences at all modems operating within the same cable. These interferences are also known as crosstalk.

Fig. 3.1 shows a typical multi-line system topology and the different kinds of existing crosstalk (exemplary for two of the three users). Interferences from a transmitter into a receiver at the same end of the cable are referred to as NEXT. The level of it increases with frequency and is independent of cable length. Near-end crosstalk gets intolerably high at frequencies used in VDSL and VDSL2 (15 MHz and higher), making it the major impairment for systems that share the same frequency band for US and DS transmission (e.g. echo-canceled systems like HDSL). Due to that reason, most practical systems avoid NEXT by applying duplexing methods (see Sec. 2.3). Frequency division duplexing eliminates NEXT by giving upstream and downstream different, non-overlapping frequency bands. When in addition all lines use the same bandwidth assignment, NEXT does not occur. Time division duplexing avoids NEXT by allowing only US or DS transmission at a time, requiring synchronization of all modems in the network. In this thesis, solely FDD-based and thus NEXT-free systems are considered.

Far-end crosstalk is the coupled signals from a transmitter to the receivers at the opposite end of the cable. In addition to frequency, the FEXT impact also depends on the line length and the network configuration. It is attenuated by traversing the full length of the victim's line and increases with mounting coupling length. As a consequence, FEXT is usually less serious than NEXT, but still the dominant impairment when no NEXT is present. The FEXT interference level can be 10 to 20 dB higher than the background noise and results in a large degradation of the maximum achievable data rate. It is worst among adjacent pairs and does not change much over time.

Due to the severity of FEXT, the DSL channel is better modeled as a multiuser channel of the additive noise form. The DSL system can be treated as a Multiple Input Multiple Output (MIMO) system, for which crosstalk is no longer modeled as noise but as an input. The DSL system is DMT-based (see Sec. 2.2.1), so that the transmission process can be described per tone when perfect synchronization is assumed. On a single tone k and for N users (i.e. N twisted pairs) in a cable binder, the receive signal $y_k^{(n)}$ of user n is given as a sum of the useful signal, the crosstalk contributions of all other users and additive noise:

$$y_k^{(n)} = \sum_{m=1}^{N} H_k^{(n,m)} \cdot x_k^{(m)} + n_k^{(n)}. \tag{3-1}$$

$x_k^{(m)}$ represents the transmit signal on line m for tone k. For $m = n$, $x_k^{(m)}$ is the useful signal transmitted on the line. $H_k^{(n,m)}$ is the transfer coefficient on tone k. For $m = n$ it describes the transmission from transmitter n to receiver n. For $n \neq m$ it describes the crosstalk impact of user m on user n. $n_k^{(n)}$ is the additive noise contribution on line n, which can contain thermal noise, impulse noise, RFI, alien crosstalk etc. It is assumed being spatially white and Gaussian, which is valid for most VDSL and VDSL2 deployments.

Equation 3-1 only considers the channel and crosstalk coefficients which describe the impact on user n. The total MIMO transmission, which inherits the mutual interferences between all users in the cable binder, can be described by the channel transfer matrix [TH00]

$$\mathbf{H}_k = \begin{bmatrix} H_k^{(1,1)} & H_k^{(1,2)} & \cdots & H_k^{(1,N)} \\ H_k^{(2,1)} & H_k^{(2,2)} & \cdots & H_k^{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ H_k^{(N,1)} & H_k^{(N,2)} & \cdots & H_k^{(N,N)} \end{bmatrix}. \tag{3-2}$$

It contains the direct channel coefficients as well as the crosstalk coefficients of all users in the system and therefore is of size $N \times N$. The diagonal elements $H_k^{(n,n)}$ of matrix $\mathbf{H}_k$ correspond to the direct channel coefficients of the different lines and describe the impact of the direct channel of user n on his transmit signal. The off-diagonal elements correspond to the crosstalk interference contributions and are the crosstalk coefficients. For the entire DSL system, the transmission is mathematically given in vectorized form by

$$\mathbf{y}_k = \mathbf{H}_k \cdot \mathbf{x}_k + \mathbf{n}_k, \tag{3-3}$$

where $\mathbf{y}_k = [y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(N)}]^T$, $\mathbf{x}_k = [x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(N)}]^T$, and $\mathbf{n}_k = [n_k^{(1)}, n_k^{(2)}, \ldots, n_k^{(N)}]^T$ are the receive, transmit and noise signal vectors, which contain the receive, transmit and noise signals of users n on tone k.

3.1 Upstream Transmission

In upstream, data is transmitted from the customer premises to the CO. The receiving modems are collocated, wherefore a crosstalk signal traveling from a disturber to a victim has to propagate through the full length of the disturber's line to get to the victim's receiver. Figure 3.2 depicts an US transmission, where Transmitter (Tx) m is the disturber and Receiver (Rx) n is suffering from his crosstalk. The insulation between the lines leads to an increased attenuation of the crosstalk signals. As a consequence, the direct channel $H_k^{(m,m)}$ of a transmitter is always larger than each crosstalk channel from this transmitter into another receiver n. In each column of $\mathbf{H}_k$, the diagonal element has the largest magnitude. Following [CGBM06], this property is called Column-Wise Diagonal Dominant (CWDD). It was validated by extensive measurement campaigns of real binders and satisfies

$$|H_k^{(n,m)}| << |H_k^{(m,m)}|, \; \forall n \neq m. \tag{3-4}$$

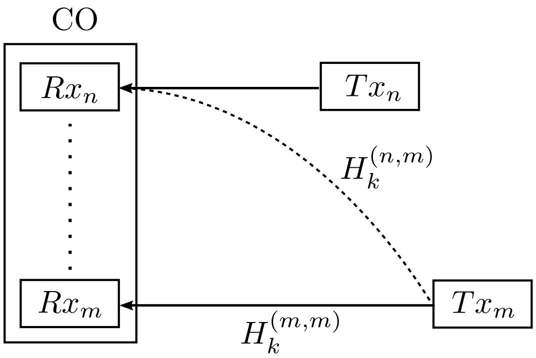


Figure 3.2: Crosstalk situation in upstream

As described, the crosstalk generated by a transmitter into the other lines of the binder is always smaller than his direct channel gain. On the contrary, the direct channel gain of a victim can be smaller than the gain of the crosstalk channels affecting this user. This effect can occur in cable binders consisting of lines of varying length with the victim line's transmitter having a larger distance to the CO than the disturber line's transmitter. Figure 3.3 depicts such kind of scenario, with line n being the victim and lines m and $m-1$ being the disturbers. The considered receiving modem can face strong signals from the closer modems

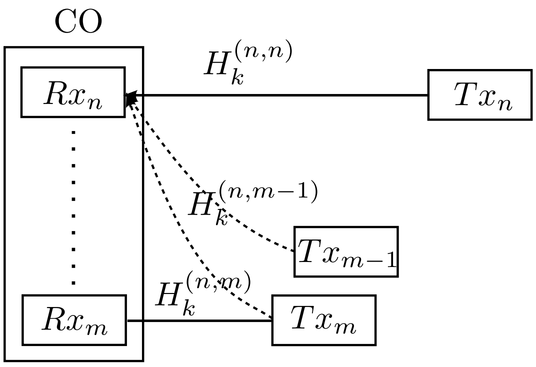


Figure 3.3: Near-far problem in upstream

as signal strength decreases with line length. It is possible that even with the attenuation of the insulation, the received crosstalk signal is stronger than the useful signal. This situation is called the near-far problem.

3.2 Downstream Transmission

In downstream, where data is transmitted from the modems in the CO to the individual customer premises, transmitters are collocated. Consequently, a crosstalk signal transmitted

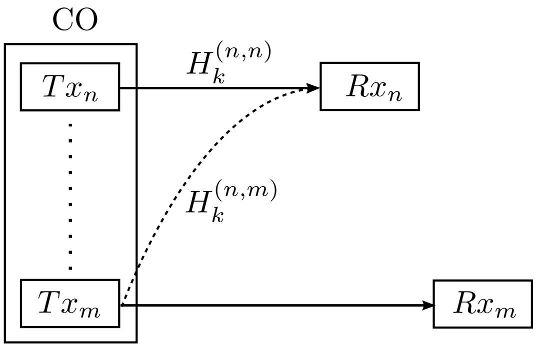


Figure 3.4: Crosstalk situation in downstream

from a disturber into a victim propagates through the entire length of the victim's line. This is shown in Fig. 3.4 with Tx m being the disturber and Rx n being the disturbed user. The useful signal and the crosstalk signals experience the same propagation length, but the crosstalk signals are additionally attenuated by the insulation. This leads to a crosstalk contribution smaller than the direct channel signal. The crosstalk channel gain of any user's transmitter into another user's receiver is always lower than the direct channel gain of that user. In each row of $\mathbf{H}_k$ the diagonal element has the largest magnitude. As described in [CGBM07], this property is called Row-Wise Diagonal Dominant (RWDD) and is fulfilled if

$$|H_k^{(n,n)}| >> |H_k^{(n,m)}|, \ \forall n \neq m \tag{3-5}$$

is valid. Like CWDD in upstream, RWDD was verified through extensive measurement campaigns of real binders.

3.3 Crosstalk Channel Modeling

The direct channel transfer function $H^{(n,n)}(f,d)$ of a twisted-pair can be accurately estimated by the RLCG model. This was shown and explained in detail in Sec. 2.4. For the estimation of the crosstalk transfer functions $H^{(n,m)}(f,d)$, $n \neq m$, there exists a variety of models which differ in their closeness to measured crosstalk channels.

Commonly used are empirical worst case models. For a cable, the standardized 99% worst-case models provide an estimate of the FEXT influence from all other lines in the binder. In 99% of the cases the practically occuring crosstalk amplitude is less than predicted by the model. Worst-case models are suitable for worst-case analysis. But they do not incorporate all channel characteristics found in measured FEXT channels.

Stochastic crosstalk channel models like for instance presented in [SDS$^+$07], consider phase coupling between the lines and variations of the crosstalk amplitude from one line to the other. They are based on the worst-case approaches, but usually model the phase coupling as uniformly distributed and the amplitude variations as Gaussian or Beta distributed. Further proposals additionally model the irregular FEXT variations over frequency by a cosine function [CFG02] or by using a sum-of-sinusoids [XSH09]. Another solution are physical models, which have a solid physical background, but a high computational complexity. They are based on cascades of many small and homogeneous binder segments and use the primary transmission line parameters R, L, C, and G [LCJ$^+$07].

In this thesis a stochastic crosstalk channel model, the so-called Beta model, is considered. It relies on measurements of real cables and is valid for TP1 and TP2 lines. Besides including variations of crosstalk amplitude and phase coupling over frequency and/or distance, it also considers the geometry of the binder. Up to 4 binders with a maximum of 25 pairs each can be simulated with this model.

The basis of the Beta model is the 99% worst-case approach, which is applied to calculate the crosstalk channel amplitudes by

$$\left|H_{99}^{(n,m)}(f,d)\right| = \left|H^{(n,n)}(f,d)\right| \cdot f \cdot \sqrt{d_{\text{coupling}}^{(n,m)}} \cdot \kappa_{FEXT},\ n \neq m, \tag{3-6}$$

with coupling factor κ_{FEXT} set to $1.594 \cdot 10^{-10}$ and coupling length $d_{\text{coupling}}^{(n,m)}$. Consequently, it complies with the 99% worst-case model and only 1% of the practically occurring crosstalk coupling is stronger than the simulated crosstalk amplitudes.

The crosstalk channel transfer functions including phase and binder geometry are calculated by

$$H^{(n,m)}(f,d) = \left|H_{99}^{(n,m)}(f,d)\right| \cdot e^{j\phi(f)} \cdot 10^{\frac{X_{dB}}{20}},\ n \neq m. \tag{3-7}$$

X_{dB} is the amplitude offset of the crosstalk transfer function in dB, relative to the 99% worst-case model. It does not depend on frequency and its probability density function is Beta distributed [MGP09]. $\phi(f)$ is the phase of the crosstalk transfer function. In this thesis it is assumed of being the same as the corresponding direct channel phase.

Figure 3.5 shows an example for crosstalk channel functions generated with the described crosstalk channel model. It shows the direct and crosstalk channel transfer functions for a user with four disturbers in a scenario with different line lengths. Three DS and two US bands can be observed. Starting at the left the first, third and fifth band are used for downstream transmission, whereas the second and fourth band is available in upstream. High variations of

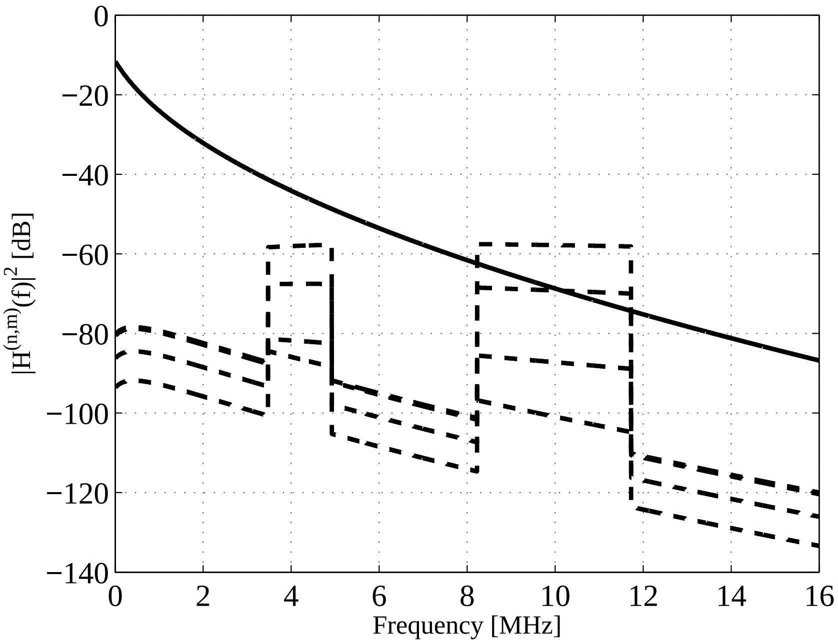


Figure 3.5: Direct channel transfer function (solid) and modeled crosstalk channel transfer functions (dashed) of an example DSL line

crosstalk strength can be seen in upstream especially for higher frequencies. The differences of crosstalk strength in downstream are below 20 dB.

IV

Crosstalk Cancellation

In a typical telephone copper cable network, the twisted copper pairs of the different users are bundled within large cable binders typically containing up to 100 individual pairs. Due to non-perfect insulation of the twisted wire pairs there is significant electromagnetic coupling among nearby lines. This leads to severe crosstalk between the users transmitting within a binder.

There exist two different kinds of crosstalk, near-end crosstalk and far-end crosstalk. Near-end crosstalk occurs when transmit signals of one stream direction disturb the receive signals of the other stream direction. It can be avoided by frequency division duplexing, where different frequency bands are applied for upstream and downstream. Interferences from transmit signals of the same stream direction, so called far-end crosstalk, strongly decrease the performance in xDSL systems. The far-end crosstalk interference level can be 10 to 20 dB larger than the background noise and therefore represents the largest performance limiter in the system. Here only far-end crosstalk is considered. When crosstalk is mentioned it is referred to FEXT.

Different kinds of proposals to reduce the influence of crosstalk have been made in literature. When collocation on both sides of the DSL link is given, joint linear processing is possible. [TH00] describes a pre- and post-filtering solution using singular value decomposition leading to increased performance. Though, simultaneous collocation at the CO and of the customer premises is generally not given.

For upstream transmission where receivers are typically centralized at the CO, several canceler designs have been proposed. [GC01] and [YC00] describe Successive Interference Cancellation (SIC) applying a Decision Feedback Canceler (DFC). A DFC is a non-linear canceler, which decodes one user at a time and uses the estimates to decode the next one. For perfect channel coding, it achieves near-optimum performance, but at the expense of infinite complexity and latency. Therefore in practical systems, where channel coding is non-perfect, SIC can suffer from error propagation, which decreases performance. A Minimum Mean Square Error (MMSE) solution is presented in [GDJ08]. It is well known to balance FEXT cancellation and noise amplification by considering the correlation characteristics of the noise in the canceler design. A crosstalk cancellation technique based on Turbo coding is presented in [DP02]. It is able to allow large performance increases but at the same time it is extremely complex.

For downstream, recursive pre-distortion by a Tomlinson-Harashima precoder was proposed in [GC00]. The THP is able to provide crosstalk-free transmission for all users, but can also suffer from decoding errors on the receiver side. In addition to this it requires a change of modems in the customer premises as a modulo-operation needs to be performed in the receiver.

In this thesis, crosstalk mitigation techniques based on linear zero-forcing architectures are considered [CGBM06, CGBM07]. They only require collocation on one side of the cable and despite their design is rather simplistic and they are low-complex, they are near-optimal in performance, as they can achieve interference-free transmission. Additionally, the zero-forcing approaches are scalable in complexity and support single user detection as well as complete crosstalk reduction, while showing a smooth performance degradation for decreasing complexity.

Section 4.1 explains the topics of full crosstalk cancellation in upstream and full crosstalk precoding in downstream. In Sec. 4.2, solutions which focus on partial elimination and avoidance of crosstalk are presented.

4.1 Full Crosstalk Cancellation and Precoding

Full crosstalk cancellation or precoding is performed, when the entire crosstalk in the cable binder is removed by signal processing. It allows all users to achieve maximum performance and relies on knowledge of all direct and crosstalk channel coefficients.

In most DSL transmission scenarios, collocation is not given at both ends of the line, as they may start in the CO, but commonly end in different customer premises. This makes it impossible to equalize all crosstalk on either end of the wire, as access to all receive signals cannot be provided. Consequently, crosstalk cancellation and crosstalk precoding are performed at that end of the cable where all lines are accessible.

Section 4.1.1 describes upstream zero-forcing crosstalk cancellation, which equalizes the signals on the receiver side. In Sec. 4.1.2 , downstream decomposition-based zero-forcing precoding is described, where the transmit signals are pre-disturbed in the transmitters.

4.1.1 Zero-Forcing Crosstalk Canceler

In upstream, receivers are collocated as data is transmitted from the different users to the CO. Crosstalk cancellation can be performed on the receiver side, which is shown in the block diagram in Fig. 4.1. Zero-forcing crosstalk cancellation simply equalizes the received signals

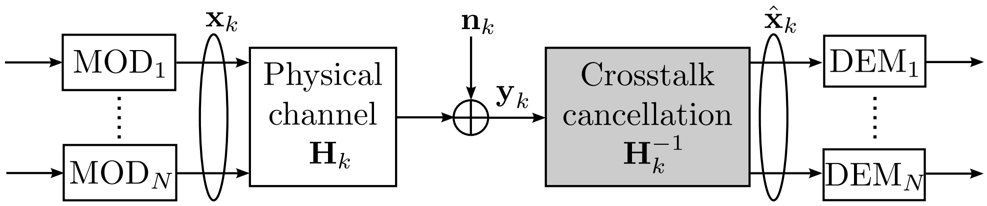


Figure 4.1: Upstream transmission block diagram with crosstalk cancellation

with the inverse of the channel matrix $\mathbf{H}_k$ for every tone k. The estimated received signal vector is given by

$$\hat{\mathbf{x}}_k = \mathbf{H}_k^{-1} \cdot \mathbf{y}_k \tag{4-1}$$

$$= \mathbf{H}_k^{-1} \cdot (\mathbf{H}_k \cdot \mathbf{x}_k + \mathbf{n}_k) \tag{4-2}$$

$$= \mathbf{x}_k + \tilde{\mathbf{n}}_k. \tag{4-3}$$

When synchronization is perfect and channel and crosstalk transfer functions are known without any inaccuracy, the received signal is only influenced by the scaled noise contribution $\tilde{\mathbf{n}}_k$ after cancellation. As upstream crosstalk cancellation removes all crosstalk completely, it also solves the near-far problem.

Zero-forcing equalization is known for its easy calculation of filter coefficients, but also for the risk of noise enhancement in ill-conditioned channels. In upstream DSL, the majority of channel matrices $\mathbf{H}_k$ is well-conditioned, based on the CWDD property explained in Sec. 3.1. In [CGBM06] it was shown that for CWDD channels, where $|H_k^{(m,m)}| >> |H^{(n,m)}|$ for $n \neq m$, the norm of the matrix inverse $||[\mathbf{H}_k^{-1}]_{\text{row } m}||^2$ is upper bounded by $||[\mathbf{H}_k^{-1}]_{\text{row } m}||^2 \leq |H_k^{(m,m)}|^{-2}$. This leads to the scaled noise PSD being upper bounded to $\tilde{\sigma}_m^2(k) \leq \sigma^2/|H_k^{(m,m)}|^2$. Since $|H_k^{(m,m)}|^2 < 1$ the noise PSD increases, but it does not get infinitely high.

Figure 4.2 shows an example for the performance gains, which can be achieved with full zero-forcing crosstalk cancellation. A huge increase in achievable data rate is visible for all line lengths.

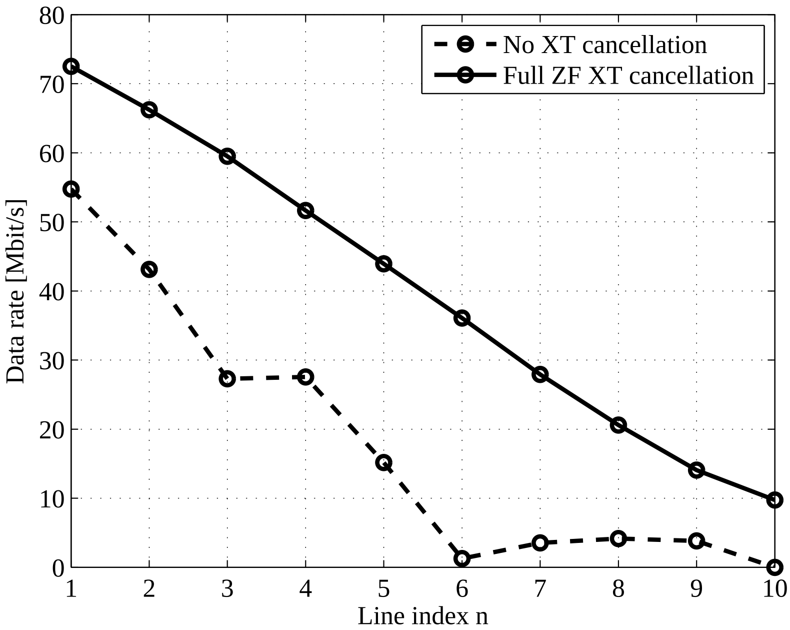


Figure 4.2: Performance increase with full upstream crosstalk cancellation

4.1.2 Decomposition-based Zero-Forcing Precoder

In downstream transmission, the different receivers have various locations and receiver coordination is not possible. On the contrary, the transmitters are collocated in the CO, which enables precoding on the transmitter side. In Fig.4.3 it can be seen that with decomposition-

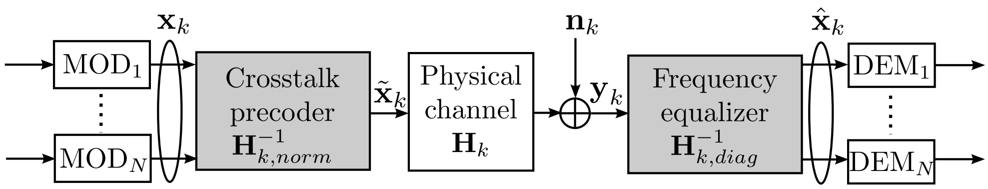


Figure 4.3: Downstream transmission block diagram with crosstalk precoding

based zero-forcing precoding, the transmit signals are pre-disturbed by the inverse of the normalized channel matrix $\mathbf{H}_{k,norm}$ for every tone. $\mathbf{H}_{k,norm}$ only contains the influence of crosstalk and is given by

$$\mathbf{H}_{k,norm} = \mathbf{H}^{-1}_{k,diag} \cdot \mathbf{H}_k, \tag{4-4}$$

where $\mathbf{H}_{k,diag} = diag\{H_k^{(1,1)}, H_k^{(2,2)}, \ldots, H_k^{(N,N)}\}$ represents a diagonal matrix containing the direct channel transfer factors. On the receiver side, only the direct channel attenuation is compensated by direct channel equalization in the individual receivers. Decomposing the channel matrix $\mathbf{H}_k$ into crosstalk-coupling and direct channel influences and precoding only with $\mathbf{H}_{k,norm}^{-1}$ avoids an undesired power increase of the transmit signal as it would occur for direct zero-forcing precoding. Due to the RWDD property of the channel, where $|H_k^{(n,m)}| << |H_k^{(n,n)}|$, $\forall n \neq m$ (see Sec. 3.2), the transmission power enhancement is $\| \; [\mathbf{H}_{k,norm}]_{\text{row } n}^{-1} \|^2 \approx 1$, meaning precoding does not lead to a significant transmit power increase [CMV+04a].

Furthermore, decomposition-based zero-forcing precoding does not suffer from poor performance like direct zero-forcing precoding [CGBM07]. It is prevented that all modems see the channel of the worst line in the binder.

Assuming that all lines are precoded by the inverse of normalized channel matrix $\mathbf{H}_{k,norm}$ before being transmitted, the receive signal $\mathbf{y}_k$ is given by

$$\mathbf{y}_k = \mathbf{H}_k \cdot \tilde{\mathbf{x}}_k + \mathbf{n}_k \tag{4-5}$$
$$= \mathbf{H}_k \cdot \mathbf{H}_{k,norm}^{-1} \cdot \mathbf{x}_k + \mathbf{n}_k \tag{4-6}$$
$$\tag{4-7}$$

with precoded transmit signal vector $\tilde{\mathbf{x}}_k$. Together with direct channel equalization on the receiver side the estimated transmit signal $\hat{\mathbf{x}}_k$ is described by

$$\hat{\mathbf{x}}_k = \mathbf{H}_{k,diag}^{-1} \cdot \mathbf{y}_k \tag{4-8}$$
$$= \mathbf{H}_{k,diag}^{-1} \cdot \left(\mathbf{H}_k \cdot \mathbf{H}_{k,norm}^{-1} \cdot \mathbf{x}_k + \mathbf{n}_k \right) \tag{4-9}$$
$$= \mathbf{x}_k + \tilde{\mathbf{n}}_k. \tag{4-10}$$

For perfect channel knowledge, crosstalk-free reception is achieved and the background noise is only enhanced by the direct channel attenuation.

Figure 4.4 shows the tremendous performance gains achievable with precoding. For all users in the binder an enormous data rate increase is visible. Due to the fact that precoding is performed at the transmitter side, the transmit signal is spectrally shaped. Example transmit spectra for the downstream tones are shown in Fig. 4.5. The initial flat spectrum before crosstalk precoding was at -60 dBm/Hz.

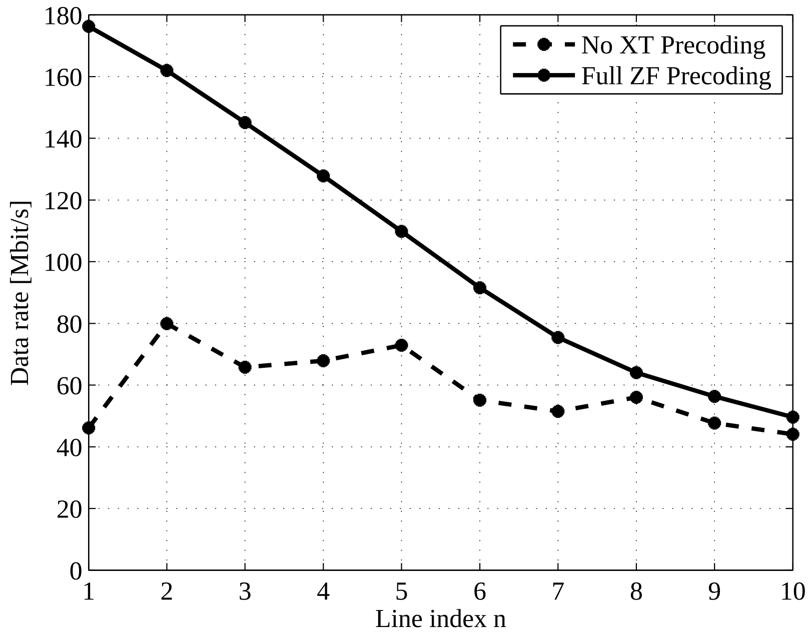


Figure 4.4: Performance increase with downstream precoding

4.2 Partial Crosstalk Cancellation and Precoding

In the last section the great benefits of crosstalk cancellation and precoding were presented. Large data rate gains can be achieved by canceling crosstalk on the receiver side in upstream or precoding the signal in the transmitter to avoid crosstalk in the receivers in downstream. The large gains of full cancellation and precoding come at the expense of high computational complexity, which can be unbearable for large binder sizes. At a symbol rate of 4 kHz, for 4096 tones and a bundle of only 10 lines transmitting on them, already 1.6 billion multiplications per second are required for full cancellation and precoding. This amount of multiplications can only be handled by large and powerful chips. Moreover, full procedures might be really challenging for cable bundles with much more lines as the number of lines influences the number of multiplications quadratically. This puts a high burden on both cost and feasibility. To enable data rate increases also for large binders and generally reduce the burden of computational complexity and therefore cost, a reduction in required computational complexity is desirable. Simultaneously, an acceptable performance still needs to be achieved.

Crosstalk selectivity is inherent in DSL systems. Major interference on a user only comes

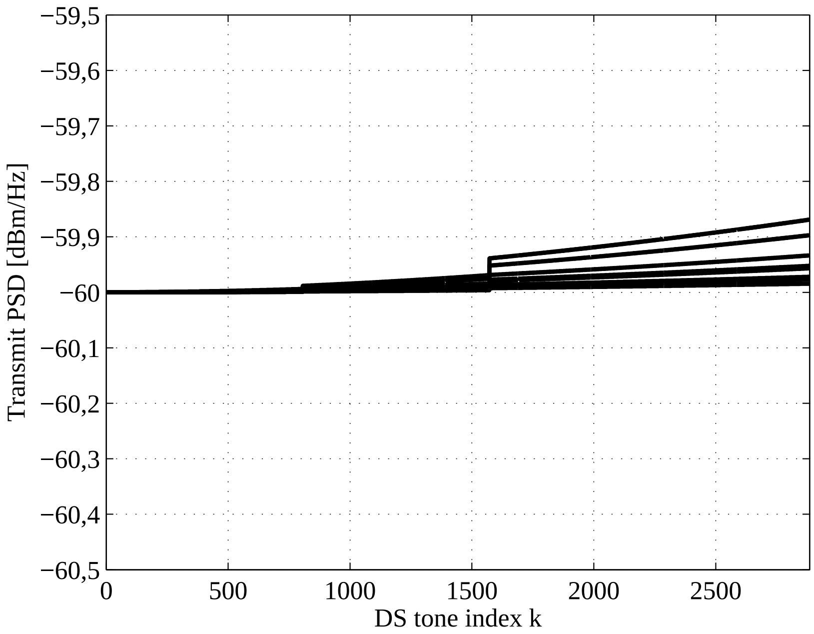


Figure 4.5: Transmit spectra for downstream precoding

from a few crosstalkers. There is space-selectivity due to the layout of the binder and frequency-selectivity due to the transmission channel. Both can be exploited for complexity reduction of crosstalk cancellation and precoding. Partial crosstalk cancellation and precoding only considers the largest crosstalkers in the cancellation and precoding procedure.

Section 4.2.1 describes crosstalk selectivity and its benefit on partial crosstalk cancellation and precoding in more detail. In Sec. 4.2.2 the principle of partial crosstalk cancellation and the structure of the partial cancellation filter are explained. Section 4.2.3 explains partial precoding and describes the structure of the partial precoding filters. Both techniques are based on the zero-forcing approach.

4.2.1 Crosstalk Selectivity

The severity of crosstalk varies both with frequency and space. Frequency selectivity describes the change of crosstalk over frequency, whereas the variations of crosstalk over the lines in a binder is named space selectivity.

Frequency selectivity is caused by frequency-dependent electromagnetic coupling, which leads to varying crosstalk channels over frequency. Space selectivity naturally arises due to the physical layout of the binders. As electromagnetic coupling decreases with distance, some

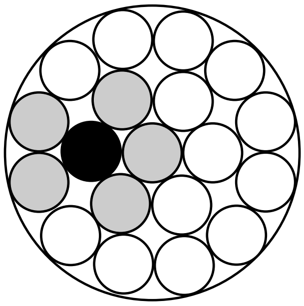


Figure 4.6: Space selectivity caused by binder geometry: user of interest (black) and dominant crosstalkers (grey)

users cause stronger and other users cause weaker interferences on a line. Thus, each user is not equally disturbed by all lines in a binder, but generally has a few strong crosstalkers. This is illustrated in Fig. 4.6.

Additionally, space-selective crosstalk channels are caused by the near-far effect. Large differences in distance from the transmitters to the receiver lead to strongly varying coupling lengths between the disturbing lines. This results in diverging crosstalk strengths making some users strong and other user weak interferers.

Exemplary measurements for both frequency- and space-selective crosstalk channels are presented in Fig. 4.7.

4.2.2 Upstream Partial Crosstalk Cancellation

4.2.2.1 Principle

This section describes the principle and design of partial zero-forcing crosstalk cancelers [CMG+04] used in upstream transmission where receivers are collocated. Like in full zero-forcing crosstalk cancellation presented in Sec. 4.1.1, the estimate of the transmitted symbols is given by a linear combination of the received signals

$$\hat{\mathbf{x}}_k = \mathbf{W}_k \cdot \mathbf{y}_k. \tag{4-11}$$

Again, $\mathbf{y}_k$ is the frequency-domain received symbol vector, but $\mathbf{W}_k$ is the partial cancellation matrix. When detecting the transmitted signal of user n the receiver only observes the direct line of user n and $p_{n,k}$ additional lines to enable crosstalk cancellation.The set of extra observation lines for user n on tone k, which is the set of considered interferers $m_{n,k}$, is defined as

$$\mathbb{M}_k^n = \{m_{n,k}(1), \ldots, m_{n,k}(p_{n,k})\}. \tag{4-12}$$

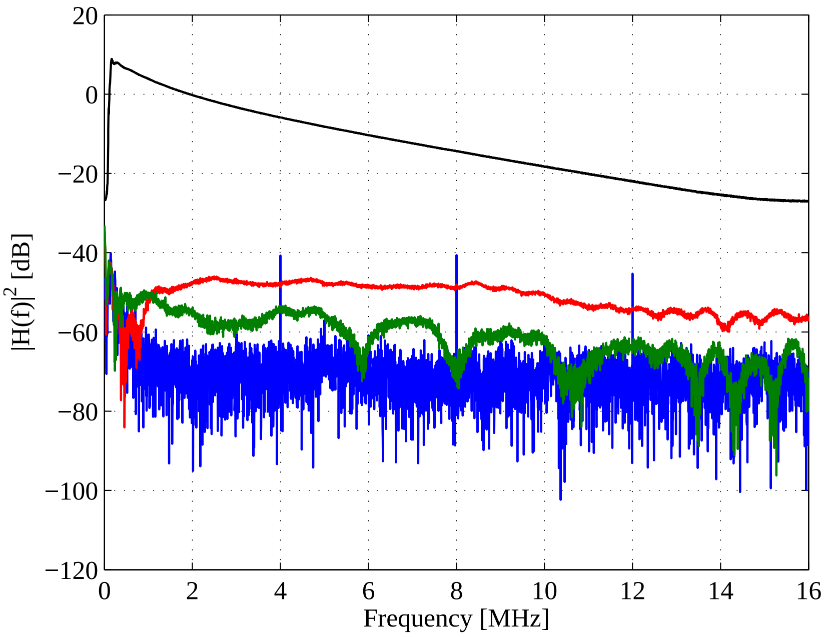


Figure 4.7: Measured direct channel and crosstalk channels

The number $p_{n,k}$ varies with user n and tone k to match the amount of crosstalk experienced by the different users on the individual tones. $p_{n,k} = 0$ corresponds to no cancellation whereas $p_{n,k} = N-1$ indicates full cancellation.

In contrary to full cancelers, partial cancelers form the estimate of the transmitted symbols by linear combinations of the received signals on the observation lines only. The received signals on the other lines are not used for the estimation of $x_k^{(n)}$. This leads to filter matrix $\mathbf{W}_k$ having a sparse structure:

$$[\mathbf{W}_k]_{n,m} = 0,\ \forall m \notin \{n, \mathbb{M}_k^n\}. \tag{4-13}$$

For the partial cancellation filter a zero-forcing design is chosen. Under the zero-forcing criterion the crosstalk from all crosstalkers contained in the set $\mathbb{M}_k^n$ is removed by the partial cancellation filter, so that

$$[\mathbf{W}_k\mathbf{H}_k]_{n,m} = \begin{cases} 1, \text{ for } m = n \\ 0, \text{ for } m \in \mathbb{M}_k^n. \end{cases} \tag{4-14}$$

For a user n, full cancellation requires $N-1$ multiplications per DMT symbol on all tones k during run-time. Due to the sparse structure of $\mathbf{W}_k$, partial cancellation on tone k only requires $p_{n,k}$ multiplications per DMT symbol. The required initialization complexity to find the selected set $\mathbb{M}_k^n$ varies with the used selection algorithm.

4.2.2.2 Partial Canceler Design

This section describes the design of the partial cancellation filter $\mathbf{W}_k$. Based on Eq. 4-11, the estimate of the transmitted signal for user n on tone k is calculated as

$$\hat{x}_k^{(n)} = [\mathbf{W}_k]_{\text{row n}} \cdot \mathbf{y}_k. \tag{4-15}$$

$\hat{x}_k^{(n)}$ is only influenced by one row of the partial canceler matrix $\mathbf{W}_k$.
In the cancellation procedure of user n on tone k only the received signals on lines $\{n, \mathbb{M}_k^n\}$ are considered as mentioned in the last section. Accordingly, the reduced vector of received signals $\bar{\mathbf{y}}_k^n$ is defined by

$$\bar{\mathbf{y}}_k^n = \left[y_k^{(n)}, y_k^{(m_{n,k}(1))}, \ldots, y_k^{(m_{n,k}(p_{n,k}))} \right]^T, \tag{4-16}$$

where the $m_{n,k}$ are given by the set $\mathbb{M}_k^n$. The reduced vector of received signals contains the signals of the considered interferers and the considered line.
In Eq. 4-13 it was shown, that the rows of the filter matrix $\mathbf{W}_k$ contain zeros for all received signals which do not participate in the partial cancellation procedure of the different users. Each row of $\mathbf{W}_k$ defines the filter coefficients for one user n, which allows to define a vector of user n's non-zero elements of $\mathbf{W}_k$ by

$$\bar{\mathbf{w}}_k^n = \left[W_k^{(n,n)}, W_k^{(n,m_{n,k}(1))}, \ldots, W_k^{(n,m_{n,k}(p_{n,k}))} \right]^T. \tag{4-17}$$

Each $W_k^{(n,m)}$ represents the element of filter matrix $\mathbf{W}_k$ in row n and column m and $W_k^{(n,m)} = [\mathbf{W}_k]_{n,m}$ holds.
With the reduced vectors the estimate of user n's symbol on tone k is given by

$$\hat{x}_k^{(n)} = \bar{\mathbf{w}}_k^n \cdot \bar{\mathbf{y}}_k^n. \tag{4-18}$$

To derive the filter coefficients, the corresponding partial channel matrix for user n on tone k is introduced as

$$\bar{\mathbf{H}}_k^n = \begin{bmatrix} H_k^{(n,n)} & [\mathbf{H}_k]_{\text{row } n, \text{ cols } \mathbb{M}_k^n} \\ [\mathbf{H}_k]_{\text{rows } \mathbb{M}_k^n, \text{ col } n} & [\mathbf{H}_k]_{\text{rows } \mathbb{M}_k^n, \text{ cols } \mathbb{M}_k^n} \end{bmatrix} \tag{4-19}$$

where $[\mathbf{Z}]_{\text{rows } \mathbb{X}, \text{cols} \mathbb{Y}}$ denotes the sub-matrix formed by the rows $\mathbb{X}$ and the columns $\mathbb{Y}$ of the matrix $\mathbf{Z}$. A partial cancellation filter based on the zero-forcing principle removes all crosstalk from crosstalkers being part of $\mathbb{M}_k^n$. It can be shown [Cen04] that the filter coefficients thus are given by

$$\bar{\mathbf{w}}_k^n = [\mathbf{I}_{p_{n,k}+1}]_{\text{col } n}^T \cdot (\bar{\mathbf{H}}_k^n)^{-1} \tag{4-20}$$

where $\mathbf{I}_{p_{n,k}+1}$ is the $(p_{n,k}+1) \times (p_{n,k}+1)$ identity matrix.

4.2.3 Downstream Partial Crosstalk Precoding

4.2.3.1 Principle

In this section the principle and design of partial zero-forcing crosstalk precoders [CGMA04] used in downstream transmission is described. In downstream, transmitters are collocated implying that crosstalk can only be avoided by a suitable pre-equalization of the transmit signal. The decomposition-based precoder design presented in Sec. 4.1.2 is chosen. It leads to the estimated transmit vector

$$\hat{\mathbf{x}}_k = \mathbf{H}_{diag,k}^{-1} \cdot \mathbf{H}_k \cdot \tilde{\mathbf{x}}_k + \tilde{\mathbf{n}}_k \tag{4-21}$$

$$= \mathbf{H}_{diag,k}^{-1} \cdot \mathbf{H}_k \cdot \mathbf{Z}_k \cdot \mathbf{x}_k + \tilde{\mathbf{n}}_k. \tag{4-22}$$

The precoded signal is formed by a linear combination of the transmit signals of the respective users

$$\tilde{\mathbf{x}}_k = \mathbf{Z}_k \cdot \mathbf{x}_k, \tag{4-23}$$

where $\mathbf{Z}_k$ is the partial precoder matrix.

When precoding the transmit signal of the users n, each user first decides on the crosstalkers disturbing him the most by selecting the set of extra observation lines

$$\mathbb{M}_k^n = \{m_{n,k}(1), \ldots, m_{n,k}(p_{n,k})\},\ \forall n,k. \tag{4-24}$$

Based on $\mathbb{M}_k^n$ the set $\mathbb{N}_k^m$ of users $n_{m,k}$ who want to be protected against interference from transmitter m is derived:

$$\mathbb{N}_k^m = \{n : m \in \mathbb{M}_k^n\} = \{n_{m,k}(1), \ldots, n_{m,k}(t_{m,k})\},\ \forall m,k. \tag{4-25}$$

$t_{m,k}$ is the number of receivers who do not want to suffer from crosstalk produced by transmitter m. All users in $\mathbb{N}_k^m$ are included in the precoding procedure. The signals on all other lines are not precoded, leading to a sparse precoding matrix

$$[\mathbf{Z}_k]_{n,m} = 0,\ \forall n \notin \{m, \mathbb{N}_k^m\}. \tag{4-26}$$

Additionally, a decomposition-based filter structure is chosen, so that

$$[\mathbf{H}_k\mathbf{Z}_k]_{n,m} = \begin{cases} H_k^{(m,m)} & \text{, for } n = m \\ 0 & \text{, for } n \in \mathbb{N}_k^m \end{cases} \tag{4-27}$$

has to be fulfilled.

4.2.3.2 Partial Precoder Design

In this section the design of the partial precoding filter $\mathbf{Z}_k$ is explained. The goal is to define a precoding filter so that the transmit signal $\tilde{x}_k^{(m)}$ for user m on tone k can be expressed as

$$\tilde{x}_k^{(m)} = [\mathbf{Z}_k]_{\text{row m}} \cdot \mathbf{x}_k. \tag{4-28}$$

Define the reduced transmit signal vector

$$\bar{\mathbf{x}}_k^m = [x_k^{(m)}, x_k^{(n_{m,k}(1))}, \ldots, x_k^{(n_{m,k}(t_{m,k}))}]^T, \tag{4-29}$$

containing the corresponding transmit signals of the considered transmitters. The vector containing the non-zero elements of $\mathbf{Z}_k$ used in the precoding of Tx m is given by

$$\bar{\mathbf{z}}_k^m = [Z_k^{(m,m)}, Z_k^{(n_{m,k}(1),m)}, \ldots, Z_k^{(n_{m,k}(t_{m,k}),m)}]^T, \tag{4-30}$$

where $Z_k^{(n,m)} = [\mathbf{Z}_k]_{n,m}$ are the elements of the filter matrix $\mathbf{Z}_k$ in row n and column m. With the reduced vectors the precoded transmit signal $\tilde{x}_k^{(m)}$ for user m on tone k is given by

$$\tilde{x}_k^{(m)} = \bar{\mathbf{z}}_k^m \cdot \bar{\mathbf{x}}_k^m. \tag{4-31}$$

In order to derive the precoding filter coefficients, the corresponding partial channel matrix for user m on tone k is introduced as

$$\bar{\mathbf{H}}_k^{pre,m} = \begin{bmatrix} H_k^{(m,m)} & [\mathbf{H}_k]_{\text{row } m, \text{ cols } \mathbb{N}_k^m} \\ [\mathbf{H}_k]_{\text{rows } \mathbb{N}_k^m, \text{ col } m} & [\mathbf{H}_k]_{\text{rows } \mathbb{N}_k^m, \text{ cols } \mathbb{N}_k^m} \end{bmatrix} \tag{4-32}$$

The filter coefficients then are given by

$$\bar{\mathbf{z}}_k^m = H_k^{(m,m)} \cdot (\bar{\mathbf{H}}_k^{pre,m})^{-1} \cdot [\mathbf{I}_{t_{m,k}+1}]_{\text{col m}}^T \tag{4-33}$$

with identity matrix $\mathbf{I}$.

V
Spectrum Management

Chapter 4 described a set of crosstalk mitigation techniques which can significantly increase the performance of a single line. Parts or all of the interferences introduced by the other lines in the binder can be eliminated by signal processing. What all these techniques have in common is, that they put a considerable burden of additional computational complexity on the reception or transmission of each symbol. This holds especially for large binder sizes. Therefore it is important to take other ways of crosstalk mitigation into account. Spectrum management is a candidate for crosstalk reduction which avoids high signal processing complexities.

DSL is a multi-user multi-carrier system. Each user transmits over multiple carriers and between all users in the binder, mutual crosstalk is generated. Consequently, the performance of a user depends on his own signal strength and the strength of the interfering signals. The user's achievable data rate depends on his own power allocation, but also on the power allocation of all other lines in the binder.

In literature, spectrum management techniques were widely discussed. They aim at avoiding unnecessary crosstalk by spectral coordination, instead of mitigating it by signal coordination and are not limited to certain transmission scenarios. Where crosstalk cancellation significantly increases the signal processing complexity of the system, spectrum management does not increase it at all.

With spectrum management the transmit spectra of the modems in a binder are limited to minimize the effects of crosstalk. For each modem the same power is available for transmission. The modem has to distribute that power to the tones in a way to achieve a trade-off between maximizing its own capacity and minimizing the crosstalk introduced on all other lines in the network. The goal is a trade-off which is fair for both the transmitter and the interfered receivers. It is especially helpful in near-far scenarios both in upstream and downstream, which are shown in Fig. 5.1.

Static Spectrum Management (SSM) (see Sec. 2.2.3) is the classical approach, where each DSL line maximizes its own performance by waterfilling algorithms, without considering

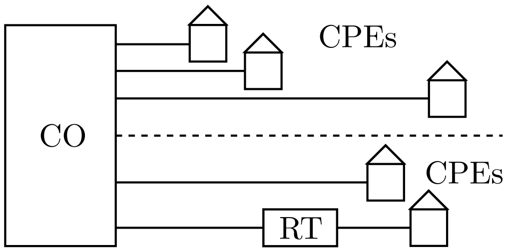


Figure 5.1: Typical scenarios to apply spectrum management techniques: Near-far scenario in US (top), Remote terminal deployment in DS (bottom)

the impact on all other lines. Examples are modems in power-adaptive mode, which aim at margin maximization for a fixed data rate and in rate-adaptive mode, where all available power is used to maximize the data rate. SSM is widely applied in DSL systems. Spectral compatibility between lines is ensured by employing spectral masks for all modems. Identical limitations on transmit power and spectra are established for all lines.

A serious disadvantage of static spectrum management is the fact that spectral masks are based on worst case crosstalk emission scenarios. This makes them overly restrictive in many cases and results in suboptimal performance and prudent spectrum usage. Dynamic Spectrum Management (DSM) overcomes the poor performance of SSM by adjusting the spectra to the individually found crosstalk situation and taking into account the influence on other neighboring lines. Instead of treating each line in isolation, DSM looks at all lines in a given cable binder as a multi-user system. The goal of dynamic spectrum management is to increase the performance of the cable binder by increasing the reach or the data rate. The data rates of the DSL network are maximized by optimally balancing the transmit spectra. The goal is to find the optimum power allocation of all users.

The objective of this chapter is to give an overview over existing DSM techniques. Several proposals were made in literature: autonomous techniques, which aim at maximization of the lines own performance, while avoiding all unnecessary crosstalk on other lines. And coordinated techniques, which maximize their own performance while minimizing the interferences to all other users. In the following sections the mathematical description of the spectrum management problem is given (Sec. 5.1) and existing solutions are presented (Sec. 5.2).

5.1 The Spectrum Management Problem

In a multi-user system, the performance of a line depends on its own power allocation and the power allocation of all other lines in the binder. Multi-user spectrum management addresses the problem of finding the optimum power allocation for all users to maximize the data rates of the entire DSL network. Accordingly, when describing the problem mathematically, the optimization criterion has constraints linking the performance of the lines in the network to each other. For N users, the spectrum management problem is defined as:

$$\max_{\mathbf{P}_1,\ldots,\mathbf{P}_N} R_N \; s.t. \; R_n \geq R_n^{(T)}, \; \forall n < N \tag{5-1}$$

$$\text{total power}: \sum_k P_k^n \leq P_{max}^n, \; \forall n \tag{5-2}$$

where $R_n^{(T)}$ denotes the target data rate of user n and R_n is the data rate on line n. The PSD of user n is defined as $\mathbf{P}_n = [P_1^n, \ldots, P_K^n]$, with K being the number of tones. In addition to the rate constraint for all other lines in the binder, the optimization is typically subject to a total power constraint P_{max}^n, limiting the total transmission power of each modem.

5.2 Existing Solutions

5.2.1 Autonomous Methods

Autonomous spectrum management techniques aim at avoiding all unnecessary crosstalk to their neighbors. They compute the power allocation of a DSL line based on its own line condition and service requirements. The reach of autonomous techniques is limited to their own lines but they notice the influence of all other lines due to noise measurements.

5.2.1.1 Power Back-Off

Classical transmission scenarios where the effects of crosstalk can be strongly reduced by a fitting power allocation are near-far scenarios in US and mixed RT/CO scenarios in DS. In these scenarios shorter lines affect the longer lines in the binder massively by crosstalk. Reducing the transmit power of the short lines to enable a better transmission on the longer lines therefore is straight forward.

The process of reducing the transmit PSD to lower the level of FEXT on highly affected twisted pairs is called Power Back-Off (PBO). There exist various methods which differently define how to select the transmit PSD of a user. What they all have in common is the need of a certain reference to decide on the adaptation of the transmitter's power allocation.

In [Sch02, Jac01] several methods for Upstream Power Back-Off (UPBO) are described, which are also applicable in downstream.

In [VMGP10] a proposal for the downstream of CO/RT-deployed networks is shown. It focuses on defining the transmit PSD of the RT in such a way that it only causes a tone dependent defined reduction of transmission bits and bit rate on the victim line.

Power back-off is able to increase the achievable bit rate on longer lines, without a change in the network infrastructure. A severe drawback is that the results strongly depend on the chosen method and the selected parameters.

5.2.1.2 Iterative Waterfilling

Iterative waterfilling is the extension of the waterfilling procedure to a multiuser environment. It relies on the single-user waterfilling solution presented in Sec. 2.2.3 and was widely discussed in literature [SCGC02, GDJ08, SSCS03].

Starting at any initial spectra, the waterfilling procedure is performed independently on each line across all lines in the DSL binder. Each user tries to selfishly maximize his own data rate (or minimize his power at a fixed data rate) while regarding the crosstalk interference from all other users as noise. This is repeated until the algorithm converges and the resulting spectra are stable. Due to the frequency-selective crosstalk channel transfer functions, Iterative Waterfilling (IWF) brings positive effects. In [YGC02, LP06, CKLC03] it was shown that IWF converges in typical DSL deployments. It is not important whether or not the users' water fillings are performed successively or in an ordered manner as the same convergence results are obtained for both cases.

A summarization of IWF is given by Algorithm 1. The behavior of IWF is illustrated in Fig. 5.2 for the 2-user case. In each iteration of the algorithm the spectra move away from each other achieving better performance step-by-step.

Iterative waterfilling is well-known for its linear complexity in number of users and frequency tones and relative simplicity. Like every other solution presented in this section it can be implemented autonomously without requiring a spectrum management center. Each line is adapted on its own without any central control. Only the total power (or rate) needs to be communicated by a centralized network entity to the modems at the CO, which means that no modems have to be replaced. Thus, huge performance gains can be achieved compared to static spectrum management.

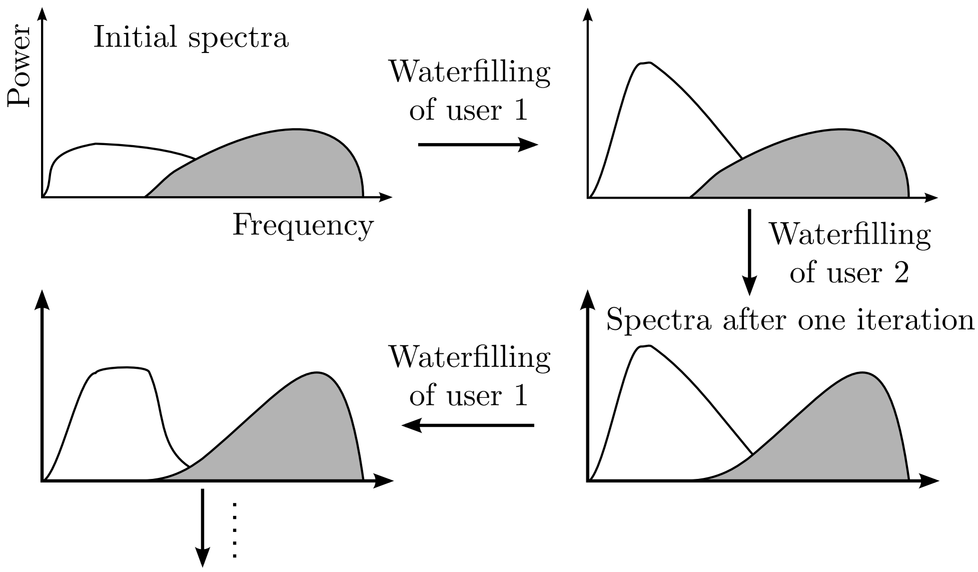


Figure 5.2: Principle of IWF for the 2-user case

5.2.1.3 Autonomous Spectrum Balancing

Autonomous Spectrum Balancing (ASB) is derived from the optimum spectrum balancing solution explained in Sec. 5.2.2. Like some power back-off algorithms it uses the concept of a reference line. For ASB, it is a representative of a typical victim line within a typical network. It is based on network statistics, not only on specific knowledge of the binder. The goal of each line is to minimize the harm it causes to the reference line, while still achieving its own data rate. In a waterfilling type of solution, ASB tries to minimize the data rate of the reference line, subject to keeping his own data rate above a target. Each user only optimizes his own PSD, as the interferences from all other users are considered as noise and the achieved rates of all other users are not of interest. This optimization is iteratively repeated for all users until their PSDs converge. As indicated by the name, ASB has no need for a centralized spectrum management center. The only information required in the modem is the direct channel, background noise and distance to the CO. The impact of a user on the reference line is determined via crosstalk channel models.

Autonomous spectrum balancing requires a low computational complexity, which increases only linearly with the number of users and tones. In [CHCM07, VBY05] it was shown that large performance gains can be achieved over existing autonomous algorithms for ADSL mixed RT/CO deployment scenarios. This is because besides the usage of a reference line, ASB differs from IWF by applying waterfilling with different waterfilling levels for different

Init: Set P_k^n to a flat PSD spectrum for all users n.
while *(No spectra convergence)* **do**
  for $n = 1 \ldots N$ **do**
    Calculate the noise spectrum:
    $\sigma_{n,total}^2(k) = \sum_{m=1, m \neq n}^{N} |H_k^{(n,m)}|^2 \cdot P_k^m + \sigma^2, \ \forall k.$
    Get P_k^n, $\forall k$ by waterfilling.
  end
end

Algorithm 1: Iterative waterfilling

tones. This allows ASB to better adapt to the special structure of the DSL channel and achieve better performance. Though it was stated in [CHCM07] that ASB is also applicable in VDSL, typical performance results for VDSL scenarios where not found in literature. For the upstream VDSL near-far scenario it was stated in [CHCM07], that convergence of the algorithms might not be given for too different line lengths.

5.2.2 Other Solutions

There exists a variety of centralized or partly centralized solutions to the spectrum management problem, which are able to outperform the autonomous solutions. They perform coordinated multi-user power allocation in order to avoid crosstalk.

Optimum Spectrum Balancing (OSB) [CYM$^+$06, CMV$^+$04b] takes into account the transmission situation of all lines in the binder and delivers the optimum power allocation of all users. However, it has an exponential complexity in the number of users, making it unusable for any DSL network with more than 5 lines.

Proposals for Iterative Spectrum Balancing (ISB) simplify OSB down to a quadratic [CM05] or a polynomial complexity [LY05], making it applicable for larger binder sizes. The approach in [VTM$^+$06] is also able to lower the complexity of OSB, but still a centralized spectrum management center is required to optimize the PSDs of all modems in the network, introducing a lot of overhead both in bandwidth and infrastructure.

Two proposals which try to reduce centralization are SCALE [PE09] and the band preference method [LKBC06]. SCALE achieves better performance than IWF at a comparable complexity, but is not autonomous as real-time messages have to be passed between users.

The band preference method optimizes the PSDs in a distributed manner, but the calculation of the band-preference coefficients again requires a spectrum management center.

VI

Crosstalk Channel Estimation

In Chapter 4, the benefits of applying crosstalk cancellation techniques have been taken into account. The limitations in data rate by crosstalk can be mitigated by zero-forcing crosstalk cancellation procedures in upstream and precoding procedures in downstream. But the increased system performance achieved by crosstalk cancellation techniques can only be obtained with full CSI and high CSI accuracy in both upstream and downstream. Therefore the direct and crosstalk channels need to be measured initially with adequate precision before any transmission including crosstalk cancellation can be done.

The DSL channel is generally stable, nevertheless it can vary slowly, for example due to temperature or humidity changes. Although these changes are relatively small and smooth, without any adaptation of the coefficients they will lead to an additional inaccuracy in the CSI knowledge and performance degradation. Accordingly, updating the canceler coefficients is needed to deploy the full potential of the DSL system.

In this thesis, channel estimation and channel update techniques are analyzed for the VDSL channels. A straightforward way to estimate and update the CSI is to periodically transmit a set of pilots as shown in [LMOW07]. These classical pilot-based estimation techniques provide good performance but have the disadvantage of utilizing parts of the useful bandwidth for pilot transmission. In addition to this there is a large signaling overhead in downstream estimation and update given by the estimates which need to be sent back to the CO. In [DMS+07, DBM+08], Least Mean Squares (LMS)-based channel estimation and adaptation algorithms are applied in order to decrease computational complexity of the adaptive precoder and canceler as well as to reduce overhead in downstream by feeding back the error samples. Further overhead reduction in downstream is obtained in [LV06b, LV06a] where the proposed estimation method only requires the sign of the error samples as channel information.

This chapter introduces channel estimation and update procedures which combine the good performance of the pilot-based estimation techniques with low pilot and signaling overhead as published by the author in [DRR10, DR10, RDR08]. In Sec. 6.1 an estimation algorithm for

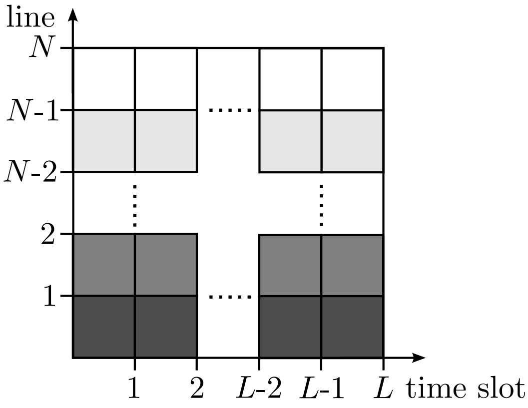


Figure 6.1: Transmission pattern for initial channel estimation

CSI is proposed which is based on orthogonal pilot sequences. For channel update it is further suggested that pilot signals are additionally transmitted during the sync frame which avoids any additional overhead in the DSL system (see Sec. 6.2). In upstream, where the receivers are collocated, a receiver-based procedure is presented. In downstream, CSI is lacking on the transmitter side and therefore a transmitter-based procedure with error feedback is applied. The simulation results for both pilot-based channel estimation and update are presented in Sec. 6.3.

6.1 Channel Estimation

There are various ways to measure the direct and the crosstalk channels of a copper cable binder. Pilot-based channel estimation methods offer high measurement accuracy and avoid any error propagation compared to decision-directed schemes. Transmitting a pilot sequence instead of just one pilot improves the measurement accuracy because estimation noise is reduced. To decrease the needed time for estimation, an orthogonal pilot-based estimation technique is applied, which allows simultaneous transmission on all lines. Various orthogonal sequences have been analyzed in literature [Lük92].

To initially estimate the direct and crosstalk channel coefficients, a set of N orthogonal pilot sequences of length L is transmitted on one line each over time (see Fig. 6.1). The transmitted pilot sequences are described by an $N \times L$ matrix $\mathbf{R}$, which is given by

$$\mathbf{R} = \begin{pmatrix} r^{(1,1)} & \dots & r^{(1,L)} \\ \vdots & \ddots & \vdots \\ r^{(N,1)} & \dots & r^{(N,L)} \end{pmatrix}.$$

Each row of matrix $\mathbf{R}$ contains L transmit pilots for one user n. For instance, for user 1 pilots $r^{(1,l)}$ with $l = 1 \dots L$ are transmitted in the first to the Lth symbol. The same pilot matrix is used for all tones k.

After reception of the complete sequence the direct and crosstalk coefficients can be calculated via correlation. In upstream and downstream each estimated direct and crosstalk coefficient is then given by [Pro01]

$$\hat{H}_k^{(n,m)} = \frac{\sum_{l=1}^{L} \tilde{r}_k^{(n,l)} \cdot r_k^{(m,l)*}}{\sum_{l=1}^{L} \left| r_k^{(m,l)} \right|^2}, \ \forall n,m \tag{6-1}$$

where $\tilde{r}_k^{(n,l)}$ is the lth received pilot on line n and tone k and $r_k^{(m,l)}$ is the lth transmitted pilot on line m. The received pilot $\tilde{r}_k^{(n,l)}$ is calculated by

$$\tilde{r}_k^{(n,l)} = \sum_{m=1}^{N} H_k^{(n,m)} \cdot r_k^{(m,l)} + n_k^{(n,l)}. \tag{6-2}$$

The signal-to-noise ratio for the channel estimation can be calculated by

$$SNR = \frac{\sigma_s^2}{\sigma^2/L} \tag{6-3}$$

with transmit signal variance σ_s^2 and noise variance σ^2, mathematically showing the increase of estimation accuracy with increased sequence length L.

6.2 Channel Update

In order to guarantee good crosstalk cancellation performance, the DSL channel does not only need to be measured once. Although DSL channels are generally stable, they can still vary slowly, e. g. due to temperature changes [Wer91, LV06b]. This requires a continuous estimation of channel parameters. Especially crosstalk channel knowledge has to be updated regularly as crosstalk cancellation techniques for upstream and downstream are very sensitive to inaccurate CSI knowledge.

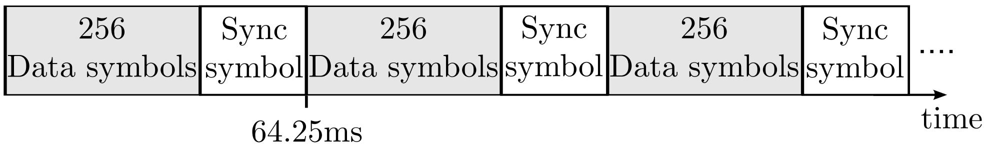


Figure 6.2: DSL frame structure

6.2.1 Pilot-based Channel Update

One goal for using a pilot-based channel update technique in the VDSL2 system is to use the existing frame structure to avoid the transmission of additional pilots instead of data symbols. During DSL transmission every 257th transmitted symbol is a sync symbol, which contains two bits as a sync flag (see Fig. 6.2). Normally, it is used for synchronization purposes but could be redefined and alienated to transmit pilot signals.

To achieve channel estimations of sufficient accuracy, pilot sequences are transmitted over time. To save estimation time and enable parallel transmission on all lines, orthogonal pilot sequences are used. In contrary to channel estimation described in the last section, pilots are not transmitted in every consecutive symbol for channel update. One pilot of the sequence is transmitted every superframe, i.e. every $T_{\text{superframe}} = 64.25\,\text{ms}$ (see Fig. 6.2). The transmission of the sequences is continuously repeated. After reception of a complete sequence, the channel coefficients can be estimated via correlation.

In upstream, collocation of the receivers is given and therefore a canceler can be applied on the receiver side. After channel estimation all CSI is available and the canceler update is done by computing $\hat{\mathbf{H}}_k^{-1}$. In downstream, receivers are not collocated wherefore transmitter-sided precoding is applied. This implies that some information has to be fed back, as full channel state information is required on the transmitter side. Signaling overhead has to be kept as small as possible. Due to standardization purposes, the normalized complex error $e_k^{(n)}(t)$ for user n and time t is fed back given by [ACA+07]

$$e_k^{(n)}(t) = \hat{r}_k^{(n)}(t) - r^{(n)}(t) \tag{6-4}$$

$$= \frac{1}{\hat{H}_k^{(n,n)}} \cdot \tilde{r}_k^{(n)}(t) - r^{(n)}(t) \tag{6-5}$$

where $\hat{r}_k^{(n)}(t)$ is the normalized received pilot at time instant t, $\tilde{r}_k^{(n)}(t)$ is the received pilot at time instant t and $r_k^{(n)}(t)$ is the transmitted pilot for user n at time instant t. It is given by the complex-valued deviation of the normalized received pilot to the transmitted pilot.

On the transmitter side where all feedback is collected, the time-dependent error matrix $\mathbf{E}_k(t)$ is then given by

$$\begin{aligned}\mathbf{E}_k(t) &= \hat{\mathbf{R}}_k(t) - \mathbf{R}_k(t) \\ &= \hat{\mathbf{H}}_{k,diag}^{-1}\left(\mathbf{H}_k \cdot \mathbf{R}_k(t) + \mathbf{N}_k(t)\right) - \mathbf{R}_k(t) \\ &= (\hat{\mathbf{H}}_{k,norm} - \mathbf{I}) \cdot \mathbf{R}_k(t) + \tilde{\mathbf{N}}_k(t) \end{aligned} \tag{6-6}$$

and is of size $N \times L$. $\mathbf{R}_k(t)$ contains the last transmitted pilot sequences for all users and $\hat{\mathbf{R}}_k(t)$ is the normalized received pilot sequence matrix. After calculating $\hat{\mathbf{R}}_k(t)$ on the transmitter side using $\mathbf{R}_k(t)$ and $\mathbf{E}_k$, $\hat{\mathbf{H}}_{k,norm}$ is estimated and the precoder update is given by the calculation of $\hat{\mathbf{H}}_{k,norm}^{-1}$.

6.2.2 Update Schemes

Provided the reception of a complete sequence there are two possible ways to update the CSI. At first, update can be done blockwise after each reception of a complete pilot sequence. In Fig. 6.3 the sequence of the blockwise estimation is exemplarily shown for a Walsh-Hadamard code of length $L = 4$. A channel update is only available every Lth superframe, but at the same time this saves computational cost. For blockwise channel update the coefficients at time instant $qN_{sf}T$ are given by

$$\hat{H}_{k,norm}^{(n,m)}(qN_{sf}T) = \frac{\sum_{l=1}^{L} \hat{r}_k^{(n)}((q-L+l)N_{sf}T) \cdot r_k^{(m,l)*}}{\sum_{l=1}^{L} \left| r_k^{(m,l)} \right|^2}, \ \forall n,m \tag{6-7}$$

with DMT symbol duration T and N_{sf} DMT symbols per superframe. Index q starts at L and increases by L with every enhancement.

Furthermore, all channel coefficients can be updated continuously by a sliding estimation. After an initial reception of a complete pilot sequence, the update is done every sync frame.

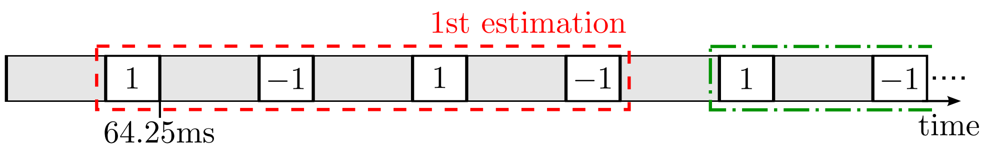


Figure 6.3: Blockwise estimation

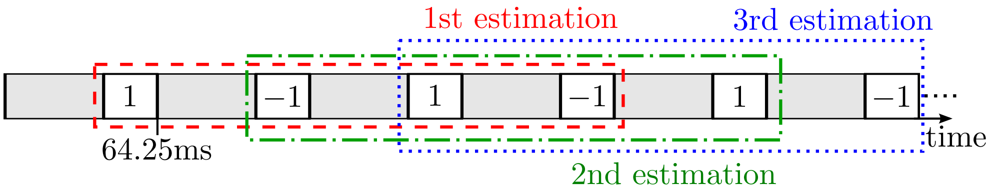


Figure 6.4: Sliding estimation

This is shown in Fig. 6.4. The advantage is the continuous update in every superframe, which comes at the expense of computational cost. For continuous update the channel coefficients at time instant $qN_{sf}T$ are given by

$$\hat{H}_{k,norm}^{(n,m)}(qN_{sf}T) = \frac{\sum_{l=1}^{L} \hat{r}_k^{(n)}((q-L+l)N_{sf}T) \cdot r_k^{(m,((q-L)modL+l-1)modL+1)^*}}{\sum_{l=1}^{L} \left| r_k^{(m,l)} \right|^2}, \ \forall n,m \qquad \text{(6-8)}$$

with DMT symbol duration T and N_{sf} DMT symbols per superframe. For the sliding estimation the index q starts at L and is increased by 1 with every raise.

6.3 Performance

6.3.1 Simulation Parameters

For the quantitative performance analysis based on simulations, several system and channel parameters need to be defined. In this section a cable binder with 10 lines of equal length is considered. The cable diameter is given by 0.5 mm (TP2) [Bri98]. No coding gain is considered, the target symbol error rate is given by 10^{-7} and the noise margin is 6 dB. All system parameters are listed in Table 6.1.

The orthogonal sequences are realized using Walsh-Hadamard codes [Lük92]. Considering the time-variance of the channel, a slow linear change of amplitude and phase of the crosstalk coefficients is assumed given by

$$\begin{aligned}
\Delta \left| H_k^{(n,m)} \right| (t) &= 20 \cdot log \left(\frac{H_k^{(n,m)}(t)}{H_k^{(n,m)}(0)} \right) = 1 \, \frac{\text{dB}}{\text{min}} \cdot t, \ n \neq m \\
\Delta \phi_{H_k^{(n,m)}}(t) &= \phi_{H_k^{(n,m)}}(t) - \phi_{H_k^{(n,m)}}(0) = 0.1 \, \frac{\text{rad}}{\text{min}} \cdot t, \ n \neq m
\end{aligned}$$

Table 6.1: Simulation parameters

Parameter	**Value**
Bandwidth B	17.664 MHz
Number of US DMT tones K_{US}	1147
Number of DS DMT tones K_{DS}	2885
Tone spacing Δf	4.3125 kHz
Coding gain γ_c	0 dB
Noise margin γ_m	6 dB
Desired symbol error rate	$< 10^{-7}$
Symbol duration	0.232 ms
Transmit signal PSD	-60 dBm/Hz
Background noise PSD	-140 dBm/Hz
Band plan	998ADE17
Shannon gap $\Gamma(P_e)$	9.8 dB
Cable type	0.5 mm
Binder size	10
Line length d	500, 700, 900, 1100 m

The direct channel coefficients remain constant. This is reasonable as inaccurate off-diagonal coefficient knowledge decreases the cancellation performance much more.

For data rate evaluation of the initial channel estimation procedure, cable lengths of $500, 700, 900$ and 1100 m are considered. Pilot sequence lengths of $L = 16$, 32 and 64 are simulated.

6.3.2 Simulation Results

6.3.2.1 Channel Estimation

Figure 6.5 and Fig. 6.6 show the performance of full crosstalk cancellation and precoding for different lengths of the pilot sequence. In upstream and downstream the achievable data rates for a binder with lines of equal length are compared to the data rate for perfect CSI.

As obvious from the figures, increasing the length of the pilot sequence decreases the measurement error and increases the data rate. For both upstream and downstream using

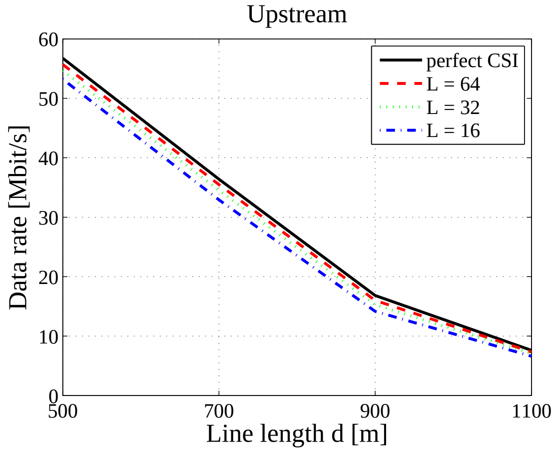


Figure 6.5: Upstream performance for different pilot sequence lengths L

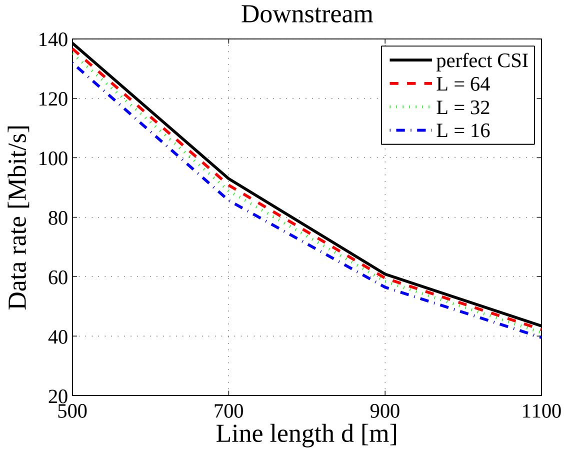


Figure 6.6: Downstream performance for different pilot sequence lengths L

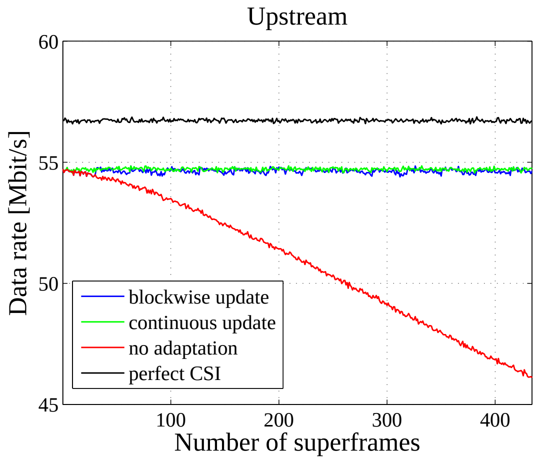


Figure 6.7: Data rate comparison in upstream

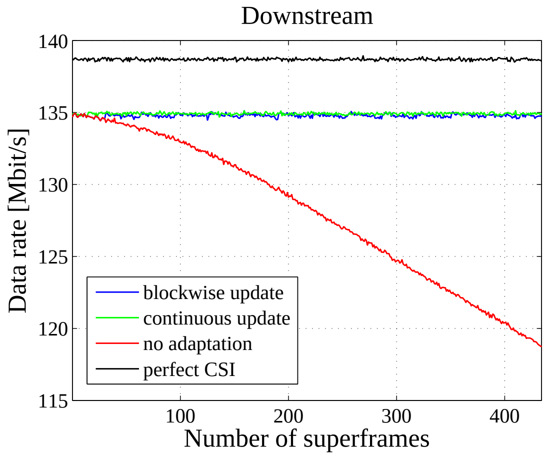


Figure 6.8: Data rate comparison in downstream

a pilot sequence of length $L = 32$ results in data rates larger than 95% of the data rate achievable with perfect CSI.

6.3.2.2 Channel Update

Figure 6.7 and Fig. 6.8 show the data rate results for adapted CSI in upstream and downstream respectively. Results are presented for a cable length of 500 m and a Walsh-Hadamard pilot sequence with length 32. The data rate over time with adapted CSI is compared to the case, where perfect channel knowledge is assumed and to the no update case. If canceler and precoder are not adapted, the initial channel estimation is necessary anyhow. As obvious from both figures, updating the CSI avoids any data rate losses due to the channel changes. Whereas the performance drops drastically within the simulation time if no update is performed, CSI update leads to a stable data rate of at least 95% of the data rate achievable with perfect CSI. It slightly drops within the updates but still leads to an acceptable performance.

6.4 Summary

In this chapter, channel estimation and update procedures are proposed and analyzed to measure the DSL channel with sufficient accuracy and to overcome data rate losses due to channel changes. The presented methods use orthogonal pilot sequences which are transmitted in an initial channel estimation as well as continuously during the sync frames. Due to transmission during the sync frames, any pilot overhead is avoided. For the channel update in upstream, correlation is applied in the receiver. In downstream, feedback of the normalized error sample is required and therefore the coefficient update is calculated on the transmitter side using correlation. It is shown that both procedures achieve high data rates by the CSI update.

In summary, the designed channel estimation procedures lead to a higher efficiency and increased performance figures in a VDSL2 system.

VII

Data Rate Constraints in DSL Access Networks

In the last two decades usage of Internet has exploded and an always increasing amount of multimedia services is provided and demanded online. High-rate applications like file sharing, HDTV and telecommuting and the wish to use multiple applications simultaneously (e. g. Triple Play) lead to a continuous aim for higher data rates. DSL providers are finding themselves in situations where users in a binder are able to reach a certain data rate, but need the guarantee for a higher one in order to run their applications stably. Consistently, DSL systems face users with different data rate constraints in the binder like shown in Fig. 7.1 a).

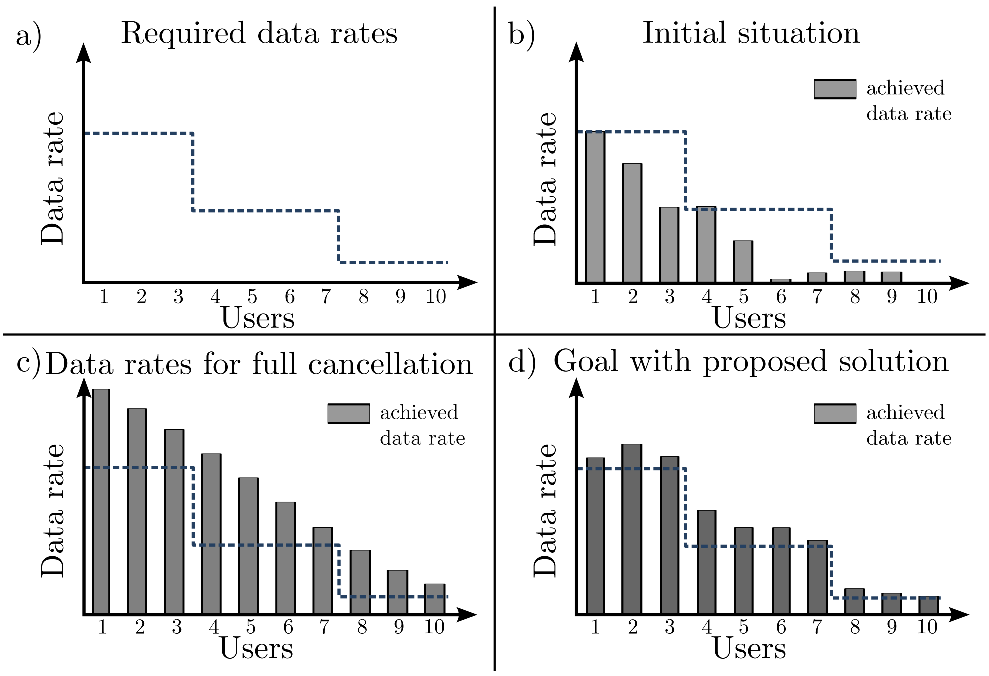


Figure 7.1: Data rate constraints in a DSL binder

When all crosstalk is present in a cable binder, the majority of users is not able to reach the required data rates for stable usage of their desired high-rate applications (see Fig. 7.1 b)). To enable all users to achieve their data rate targets, the providers need to spend additional effort on their connections.

Generally, there are two ways for a service provider to improve the quality of their DSL links: they can adapt the transmission power by spectrum management or they enlarge the amount of signal processing available on each line for crosstalk cancellation. Like presented in Chapter 5, spectrum management is able to improve the signal quality on a line but it has limitations in performance and flexibility as crosstalk-free transmission is never possible. On the contrary, signal processing techniques theoretically allow transmission without any crosstalk, but available computational complexity is bounded by feasibility and cost. In addition, full crosstalk cancellation might widely exceed the desired data rates (see 7.1 c)). Partial crosstalk cancellation techniques like presented in Sec. 4.2 could be a solution, if well-known selection algorithms would not aim at maximization of the binder capacity. As they do not focus on data rate fulfillment on the individual lines, this might increase needed computational resources. Following, providers find themselves in a trade-off between satisfying users needs and sticking to cost limits.

The objective of this chapter is to find algorithms, which guarantee the achievement of high data rate targets to all users in a DSL cable binder, but at the same time aim at minimizing the computational complexity required for that by not largely exceeded the required data rates. (see Fig. 7.1 d)). Partial crosstalk cancellation procedures with novel selection algorithms are presented, which are able to satisfy users' data rate demands at a limited amount of computational complexity. To use the available computational complexity more efficiently, the novel selection algorithms are combined with spectrum management techniques.

Chapter 7 is structured as follows: partial crosstalk cancellation techniques for data rate fulfillment are presented in Sec. 7.1. A joint partial cancellation and spectrum management approach, which aims at further reduction of needed computational complexity is proposed in Sec. 7.2. A summary of the results is given in Sec. 7.3.

7.1 Partial Crosstalk Cancellation

One of the main objectives of this thesis was to find crosstalk cancellation procedures for DSL systems, which allow all users in a binder to fulfill their data rate requirements. The focus was on adapting to the users' individual needs instead of just maximizing the data rate of the binder, to keep the required computational complexity low.

In this chapter, methods and algorithms are introduced which fulfill the data rate requirements of users in a binder with the help of signal processing. In upstream where receivers are collocated, crosstalk is reduced by a partial canceler. Downstream transmission exploits the collocation of the transmitters by avoiding crosstalk under application of a partial precoder. The basis of both the canceler and precoder is the zero-forcing structure explained in Chapter 4 as it combines good performance with a low complexity.

For signal processing, a limited computational complexity is assumed. Full cancellation procedures would exceed this complexity and would not be able to fulfill the desired rates with real-time processing. To handle data rate requirements with that limited amount of computational resources, only the complexity which is really needed can be spent on the lines. Consequently, crosstalk is not completely, but partially canceled or precoded.

A subset of interferers is observed for each user on all tones, which are chosen according to a predefined and specially tailored selection algorithm. The author has published several proposals in literature [RDR09, RDR11, DMR12]. The developed selection algorithms take into account the users' data rate requirements and distribute the available computational complexity in an intelligent way. The available computational complexity is shifted to the users who have not yet reached their data rate goals and who need to spend more additional effort on cancellation or precoding. The interferers considered in the cancellation and precoding are successively selected.

The remaining chapter is structured as follows: Sec. 7.1.1 and Sec. 7.1.2 specify the partial canceler structure in upstream and the partial precoder structure in downstream for partial crosstalk cancellation with data rate constraints. The successive crosstalk selection algorithms, which choose the canceled crosstalkers based on the users' data rate requirements are further described in Sec. 7.1.3. A complexity analysis of the solutions is carried out in Sec. 7.1.4. Performance results are given in Sec. 7.1.5.

7.1.1 UPSTREAM

To achieve given data rate constraints in upstream transmission, a partial crosstalk canceler with cancellation matrix $\mathbf{W}_{QoS,k}$ is required. The canceler eliminates crosstalk on the receiver side taking into account the data rate demands. The partial canceler structure resembles the one introduced in Sec. 4.2.2, so that the estimated transmit signal vector on tone k is given by

$$\hat{\mathbf{x}}_k = \mathbf{W}_{QoS,k} \cdot \mathbf{y}_k. \tag{7-1}$$

The off-diagonal coefficients of the partial canceler matrix have the property $W_{QoS,k}^{(n,m)} = 0$, when user m is not canceled on tone k. Each user's receive signal can be calculated by

$$\begin{aligned} \hat{x}_k^{(n)} =& \bar{\mathbf{w}}_{QoS,k}^n \cdot \bar{\mathbf{y}}_k^n \\ =& [\mathbf{I}_{p_{n,k}+1}]_{\text{col } n}^T \cdot \begin{bmatrix} H_k^{(n,n)} & [\mathbf{H}_k]_{\text{row } n, \text{ cols } \mathbb{M}_{QoS,k}^n} \\ [\mathbf{H}_k]_{\text{rows } \mathbb{M}_{QoS,k}^n, \text{ col } n} & [\mathbf{H}_k]_{\text{rows } \mathbb{M}_{QoS,k}^n, \text{ cols } \mathbb{M}_{QoS,k}^n} \end{bmatrix}^{-1} \cdot \bar{\mathbf{y}}_k^n \end{aligned} \tag{7-2}$$

where $\mathbf{I}_{p_{n,k}+1}$ is the $(p_{n,k}+1) \times (p_{n,k}+1)$ identity matrix, $\bar{\mathbf{y}}_k^n$ is the reduced received signal vector of length $p_{n,k}+1$ and $\bar{\mathbf{w}}_{QoS,k}^n$ is the vector of user n's non-zero elements of $\mathbf{W}_{QoS,k}$.

The canceler coefficients $W_{QoS,k}^{(n,m)}$ have to be chosen in a way to enable all users n to fulfill their data rate targets for a given complexity limit of C_{run}. The procedure starts with calculating the data rate difference between the target data rate $R_n^{(T)}$ and the initial data rate $R_n^{(I)}$. The initial data rate $R_n^{(I)}$ is the data rate which can be achieved without any crosstalk cancellation. It is calculated by

$$R_n^{(I)} = f_S \cdot \sum_k \log_2 \left(1 + \frac{1}{\Gamma} \cdot \frac{\left|H_k^{(n,n)}\right|^2 \cdot P_k^n}{\sum\limits_{m,m \neq n} \left|H_k^{(n,m)}\right|^2 \cdot P_k^m + \sigma^2} \right) \tag{7-3}$$

with the SNR gap Γ and the power of the transmit signal P_k^n of user n on tone k. All users n who do not yet achieve their data rate target need to cancel crosstalkers for target fulfillment.

Partial cancellation for data rate fulfillment only cancels crosstalk for user n on tone k, which is produced by the interferers contained in the set $\mathbb{M}_{QoS,k}^n$. These disturbances need to be eliminated for target data rate achievement. The set $\mathbb{M}_{QoS,k}^n$ for all users n on all tones k is determined by a selection algorithm, which takes into account the data rate requirements of the respective users. This is achieved by successively increasing the number of considered crosstalkers for the users n. Several successive selection algorithms are described in Sec. 7.1.3.

7.1.2 DOWNSTREAM

Partial crosstalk precoding for data rate fulfillment uses a transmitter-sided precoder with precoder matrix $\mathbf{Z}_{QoS,k}$, which precompensates for crosstalk on the receiver side. The precoded transmit signal vector $\tilde{\mathbf{x}}_k$ is calculated by

$$\tilde{\mathbf{x}}_k = \mathbf{Z}_{QoS,k} \cdot \mathbf{x}_k. \tag{7-4}$$

The choice of the precoder coefficients has to allow all users n to achieve their data rate goals. As already presented in Sec. 4.2.3, each precoded transmit signal can be expressed as

$$\begin{aligned} \tilde{x}_k^{(m)} =& \bar{\mathbf{z}}_{QoS,k}^m \cdot \bar{\mathbf{x}}_k^m \\ =& H_k^{(m,m)} \cdot \begin{bmatrix} H_k^{(m,m)} & [\mathbf{H}_k]_{\text{row } m, \text{ cols } \mathbb{N}_{QoS,k}^m} \\ [\mathbf{H}_k]_{\text{rows } \mathbb{N}_{QoS,k}^m, \text{ col } m} & [\mathbf{H}_k]_{\text{rows } \mathbb{N}_{QoS,k}^m, \text{ cols } \mathbb{N}_{QoS,k}^m} \end{bmatrix}^{-1} \cdot [\mathbf{I}_{t_{m,k}+1}]_{\text{col m}}^T \cdot \bar{\mathbf{x}}_k^m \end{aligned} \tag{7-5}$$

where $\bar{\mathbf{x}}_k^m$ is the reduced transmit signal vector of length $t_{m,k}$, $\mathbf{I}_{t_{m,k}+1}$ is the $(t_{m,k}+1) \times (t_{m,k}+1)$ identity matrix and $\bar{\mathbf{z}}_{QoS,k}^m$ is the vector of user m's non-zero elements of $\mathbf{Z}_{QoS,k}$.

The first step in the procedure is to calculate the difference between the initial data rate $R_n^{(I)}$, which is achieved without any crosstalk precoding, and the targeted data rate $R_n^{(T)}$. The initial data rate is given by

$$R_n^{(I)} = f_S \cdot \sum_k \log_2 \left(1 + \frac{1}{\Gamma} \cdot \frac{\left| H_k^{(n,n)} \right|^2 \cdot P_k^n}{\sum\limits_{m,m \neq n} \left| H_k^{(n,m)} \right|^2 \cdot P_k^m + \sigma^2} \right) \tag{7-6}$$

with SNR gap Γ and the power of the transmit signal P_k^n of user n. The users who have not yet achieved their data rate targets are obtained as they need to precode for crosstalk to reach their desired data rates.

The procedure only partially avoids crosstalk. The set of considered crosstalkers $\mathbb{M}_{QoS,k}^n$ is successively selected by a selection algorithm that takes into account the data rate requirements. Analogous as presented in Sec. 4.2.3, the set of receivers user m does not want to interfere on tone k, is determined out of the set $\mathbb{M}_{QoS,k}^n$ of disturbers who have to be canceled by

$$\mathbb{N}_{QoS,k}^m = \{ n : m \in \mathbb{M}_{QoS,k}^n \}, \ \forall m,k. \tag{7-7}$$

7.1.3 Successive Crosstalk Selection Algorithms

In this section, the newly developed selection algorithms are presented. They can be applied in upstream and in downstream. For a binder with data rate requirements for all users, they distribute a limited computational complexity C_{run} amongst the lines. The complexity limit C_{run} can range from $C_{run} = 0$ for no cancellation to $C_{run} = N$ (users) $\cdot (N-1)$ (crosstalkers) $\cdot$ K_u (tones) for full cancellation on all lines. K_u is the number of used tones. It equals K_{US} for upstream transmission and K_{DS} for downstream transmission. When $C_{run} = 1$ one crosstalker-tone pair can be canceled. In both stream directions, the goal for the proposed selection algorithms is to optimally use the given computational resources to first fulfill the desired data rates for all users at a minimum computational cost and then apply the remaining computational resources so that binder capacity is maximized.

Conventional selection algorithms such as those proposed in [CMG+04, CGMA04], pursue capacity maximization based on a predefined computational complexity limit for each user. In contrary, the successive crosstalk selection algorithms proposed in this section only have an overall complexity limit and strive to fulfill the different data rate requirements of the individual lines by distributing the available complexity amongst users. There is no given complexity limit for each single user, except full cancellation. The amount of computational complexity spent on each user is successively increased until the users reach their target data rates or the total available complexity is used up. Only when all users achieved their required data rates and there is still complexity to be spent, they focus on maximization of binder capacity.

The proposed algorithms work iteratively over all users. As only a predefined small fraction of the total computational complexity is given to each user in every iteration, they limit privileging of certain users during allocation of the computational resources.

In the following sections, three different successive crosstalk selection procedures are suggested. Comparable to the conventional solutions they exploit space and/or frequency selectivity by selection of the crosstalkers, the tones or the crosstalker-tone pairs which are canceled. Successive Line Selection (S-LS), Successive Tone Selection (S-TS) and Successive Joint Tone-Line Selection (S-JTLS) are described in more detail in Sec. 7.1.3.1, 7.1.3.2 and 7.1.3.3.

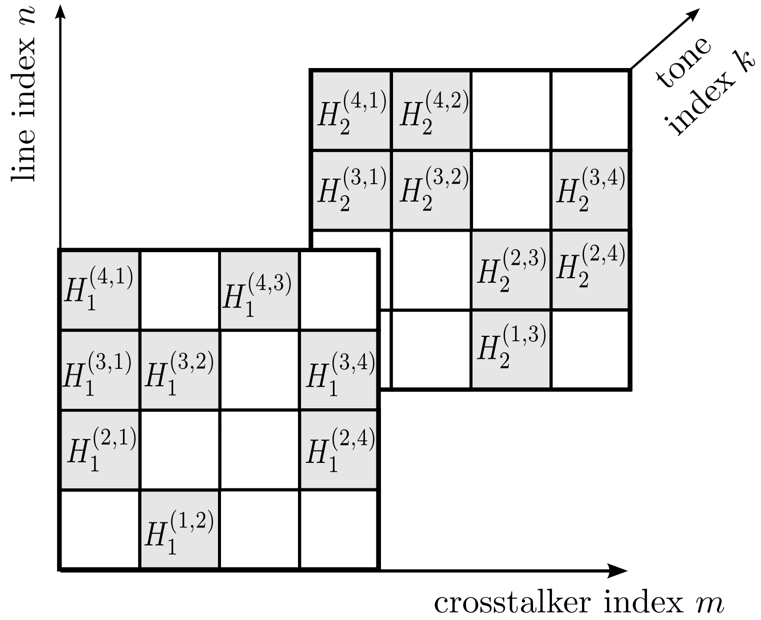


Figure 7.2: Example result of S-LS

7.1.3.1 Successive Line Selection

For each user, S-LS decides on the crosstalkers who need to be canceled in the user's cancellation procedure to fulfill his data rate requirement. S-LS cancels a constant number of interferers on each user's tones, but to satisfy the data rate requirements of all lines the number of canceled interferers can vary from one line to the other. This is shown for an easy example in Fig. 7.2, in which the colored squares indicate a canceled line. For a binder with four users and two tones, S-LS cancels one crosstalker per tone for user 1, two crosstalkers per tone for user 2, three crosstalkers per tone for user 3 and two crosstalkers per tone for user 4.

For a given complexity limit C_{run}, S-LS is able to cancel

$$c_{max} = \left\lfloor \frac{C_{run}}{K_u} \right\rfloor \tag{7-8}$$

lines for each tone in the entire binder. It distributes the complexity to the users, so that the needed complexity to fulfill all data rate requirements is the sum of canceled crosstalkers per line multiplied by the number of tones K_u.

The pseudo code for successive line selection is given in Algorithm 2 and is explained in the following. To determine the strongest interferers of each user n on each tone k, the

Init: Sort crosstalkers $m_{n,k}(i)$, $\forall i,k$ in descending order for all users n with $R_n^{(I)} < R_n^{(T)}$.
$R_{n,LS}^{(C)}(0) = R_n^{(I)}$, $\forall n$, $c_n = 0$, $\forall n$

while *($\sum_n c_n < c_{max}$ and $R_n^{(T)} > R_{n,LS}^{(C)}$, for at least one user n)* **do**
  for $n = 1 \dots N$ **do**
    if $R_n^{(T)} > R_{n,LS}^{(C)}(c_n)$ **then**
      $c_n \leftarrow c_n + 1$
      if $\sum_n c_n \leq c_{max}$ *and* $c_n \leq (N-1)$ **then**
        Calculate $R_{n,LS}^{(C)}(c_n)$
      end
      else
        $c_n \leftarrow c_n - 1$
        Exit while loop.
      end
    end
  end
end
if *($\sum_n c_n < c_{max}$)* **then**
  $\Delta c = c_{max} - \sum_n c_n$.
  Sort all remaining crosstalkers $m_{n,k}$ for all users on all tones based on their signal strength in a descending order.
  Increase c_n of the users with the strongest crosstalkers until $\Delta c = 0$.
end

Algorithm 2: Successive line selection

crosstalkers $m_{n,k}$ are sorted in descending order based on their signal strength:

$$\begin{aligned} & \{m_{n,k}(1), \dots, m_{n,k}(N-1)\} \\ \text{subject to}: & \left|H_k^{(n,m_{n,k}(i))}\right|^2 \cdot P_k^{m_{n,k}(i)} \geq \left|H_k^{(n,m_{n,k}(i+1))}\right|^2 \cdot P_k^{m_{n,k}(i+1)},\ \forall i \\ & m_{n,k}(i) \neq n,\ \forall i. \end{aligned} \tag{7-9}$$

$m_{n,k}(i)$ denotes the ith largest crosstalker of user n on tone k. For S-LS, c_n equals the number of crosstalkers that can currently be canceled for user n on all tones. Based on the sorted set of crosstalkers $\{m_{n,k}(1), \ldots, m_{n,k}(N-1)\}$, the number of crosstalkers c_n canceled for every tone of user n is increased by one in each iteration. Starting with $c_n = 0$, it is distributed to the users who did not yet achieve their data rate goals until either all of them reached their data rate requirements or the computational complexity C_{run} is used up.

After each increase of c_n, the achievement of data rate requirements is checked for the currently selected crosstalkers by calculating the current data rate $R^{(C)}_{n,LS}(c_n)$ for each user. In [RDR09, DMR12] we proposed to calculate the current data rate by exact calculation of $SINR^n_k$, which is the Signal-to-Interference and Noise Ratio (SINR) for user n on tone k. This involves matrix inversion, which is well known to be very complex in computation. As the current data rate has to be calculated in every iteration, we proposed in [DMR12] to apply an approximation for $SINR^n_k$ instead, in order to reduce the initialization complexity of the algorithm. With the approximated $SINR^n_k$ the current data rate is given by

$$R^{(C)}_{n,LS}(c_n) = f_S \cdot \sum_k \log_2 \left(1 + \frac{1}{\Gamma} \cdot \frac{\left| H_k^{(n,n)} \right|^2 \cdot P_k^n}{\sum\limits_{i=c_n+1}^{N-1} \left| H_k^{(n,m_{n,k}(i))} \right|^2 \cdot P_k^{m_{n,k}(i)} + \sigma^2} \right). \tag{7-10}$$

Equation 7-10 considers the crosstalkers that are not canceled in the denominator. To obtain whether data rate goals were reached or not, $R^{(C)}_{n,LS}(c_n)$ is compared to the data rate target $R^{(T)}_n$. When all users achieved their data rate requirement or the complexity limit C_{run} is reached, the algorithm exits the loop. The remaining complexity is given to the users which have the strongest not canceled interferers to best utilize the left-over complexity.

When S-LS was run, the selected set for cancellation for each user n is

$$\mathbb{M}^{n,LS}_{QoS,k} = \{m_{n,k}(1), \ldots, m_{n,k}(c_n)\}, \tag{7-11}$$

with the user-specific number of canceled crosstalkers c_n.

7.1.3.2 Successive Tone Selection

S-TS exploits frequency selectivity and thus allows each user to perform full crosstalk cancellation on a certain number of tones. The amount of fully canceled tones can be different from one line to the other, depending on the data rate requirement of that user. This is illustrated in Fig. 7.3 for an easy example, in which canceled crosstalkers are indicated by a colored square. For four lines and two tones, S-TS allows user 1 and user 3 to fully

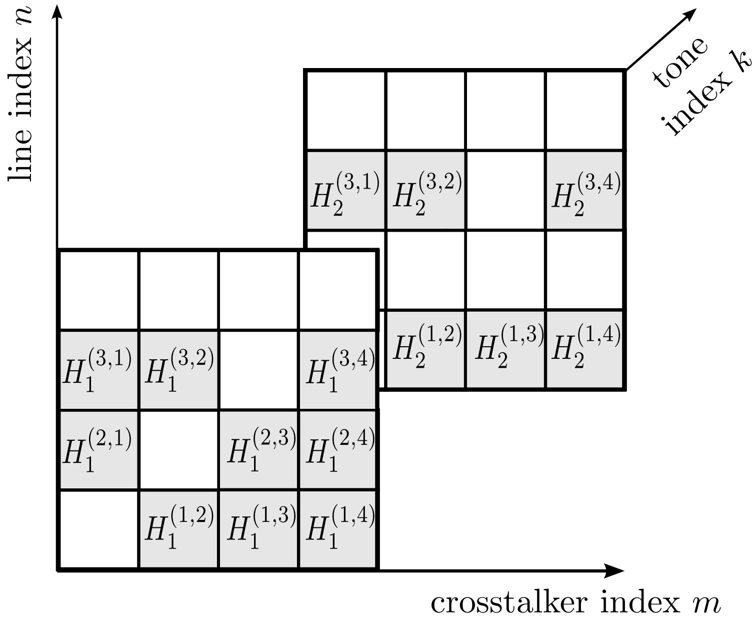


Figure 7.3: Example result of S-TS

cancel two tones. User 2 can perform full cancellation on one tone. For user 4 no crosstalk is canceled.

For a given complexity limit C_{run}, S-TS can maximally do a full cancellation for

$$t_{max} = \left\lfloor \frac{C_{run}}{N-1} \right\rfloor \tag{7-12}$$

tones in the entire binder. It distributes the available tones for cancellation to the different users based on their data rate requirements, so that the needed run-time complexity of the binder is given by the sum of canceled tones t_n for each user multiplied by $(N-1)$.

The pseudo code for S-TS is given in Algorithm 3. The algorithm starts with calculating the bit gain of full cancellation for all users n on all tones k. The bit gain $g_{n,k}^{TS}$ is calculated by

$$g_{n,k}^{TS} = log_2\left(1+\frac{1}{\Gamma}\cdot\frac{\left|H_k^{(n,n)}\right|^2\cdot P_k^n}{\sigma^2}\right) - log_2\left(1+\frac{1}{\Gamma}\cdot\frac{\left|H_k^{(n,n)}\right|^2\cdot P_k^n}{\sum\limits_{n,n\neq m}\left|H_k^{(n,m)}\right|^2\cdot P_k^m+\sigma^2}\right). \tag{7-13}$$

It is given by the number of transmitted bits for full cancellation minus the number of transmitted bits in case of no cancellation. Based on their bit gain, the tones are sorted

Init: Calculate bit gains $g_{n,k}^{TS}$, $\forall k$ for each user n.
Sort $g_{n,k}^{TS}$ in descending order for each n.
$R_{n,TS}^{(C)} = R_n^{(I)}$, $j = 0$, $t_n = 0$, $\forall n$

while *(*$\sum_n t_n < t_{max}$ *and* $R_n^{(T)} > R_{n,TS}^{(C)}$ *, for at least one user n)* **do**
  $j \leftarrow j+1$
  for $n = 1 \ldots N$ **do**
    if $R_n^{(T)} > R_{n,TS}^{(C)}(t_n)$ **then**
      Increase t_n: $t_n \leftarrow \left\lfloor \frac{j \cdot \Delta}{(N-1)} \right\rfloor$
      if $\sum_n t_n \leq t_{max}$ *and* $t_n \leq K_u$ **then**
        Calculate $R_{n,TS}^{(C)}(t_n)$
      end
    else
      $t_n \leftarrow \left\lfloor \frac{(j-1) \cdot \Delta}{(N-1)} \right\rfloor$
    end
  end
end

if *(*$\sum_n t_n < t_{max}$*)* **then**
  $\Delta t = t_{max} - \sum_n t_n$.
  Sort $g_{n,k}^{TS}$ of all uncanceled tones of all users in descending order.
  Increase t_n of the users with the highest gains until $\Delta t = 0$.
end

Algorithm 3: Successive tone selection

decreasingly:

$$\{k_n(1), \ldots, k_n(K_u)\}$$
$$\text{subject to} : g_{n,k_n(i)}^{TS} \geq g_{n,k_n(i+1)}^{TS}, \ \forall i$$

$k_n(i)$ denotes the tone with the ith largest bit gain for user n.
In the next steps, the tones which should be fully canceled are successively selected. Until the data rate targets are fulfilled, C_{run} is reached or all tones of user n are considered in the

cancellation procedure, the number of fully canceled tones t_n increases. Starting at zero, it is enhanced to

$$t_n = \left\lfloor \frac{j \cdot \Delta}{N-1} \right\rfloor \tag{7-14}$$

in every iteration of the outer loop in Algorithm 3. It depends on running index j and the complexity parameter Δ, which can be varied and defines the amount of complexity increase in every loop iteration. After each iteration the currently available data rate $R^{(C)}_{n,TS}$ is calculated by

$$\begin{aligned} R^{(C)}_{n,TS}(t_n) =& f_S \cdot \sum_{i=1}^{t_n} \log_2 \left(1 + \frac{1}{\Gamma} \cdot \frac{\left| H^{(n,n)}_{k_n(i)} \right|^2 \cdot P^n_{k_n(i)}}{\sigma^2} \right) \\ &+ f_S \cdot \sum_{i=t_n+1}^{K_u} \log_2 \left(1 + \frac{1}{\Gamma} \cdot \frac{\left| H^{(n,n)}_{k_n(i)} \right|^2 \cdot P^n_{k_n(i)}}{\sum\limits_{n,n \neq m} \left| H_k^{(n,m_{n,k_n(i)})} \right|^2 \cdot P^m_{k_n(i)} + \sigma^2} \right) \end{aligned} \tag{7-15}$$

and compared to the data rate targets of the users. Again, $SINR^n_k$ is approximated and not exactly calculated to reduce computational complexity. When the algorithm left the while-loop, the remaining complexity is distributed to the users with highest gains.

The selected set $\mathbb{M}^{n,TS}_{Qos,k}$ of tones to be fully canceled is then given by

$$\mathbb{M}^{n,TS}_{QoS,k} = \begin{cases} \{m_{n,k} | m = 1,\ldots,N \wedge m \neq n, & k \in \{k_n(1),\ldots,k_n(t_n)\} \\ \varnothing, & \text{otherwise} \end{cases} \tag{7-16}$$

for user n on tone k.

7.1.3.3 Successive Joint Tone-Line Selection

To exploit spatial as well as frequency selectivity, S-JTLS builds crosstalker-tone pairs for each user which can be selected for the cancellation procedure. The crosstalker-tone pairs are denoted as $d_n(i) = (m_n(i), k_n(i))$. $m_n(i)$ represents the crosstalker and $k_n(i)$ the tone belonging to the i^{th} crosstalker-tone pair of user n. The number of pairs canceled for a user, depends on his data rate requirements and can therefore vary from one line to the other. This is shown in Fig. 7.4 for a scenario with four lines and two tones, where a canceled crosstalker-tone pair is indicated by a colored square. S-JTLS allows user 1 to cancel two crosstalker-tone pairs. For user 2 five crosstalker-tone pairs are selected. User 3 can cancel six and user 4 can cancel three pairs.

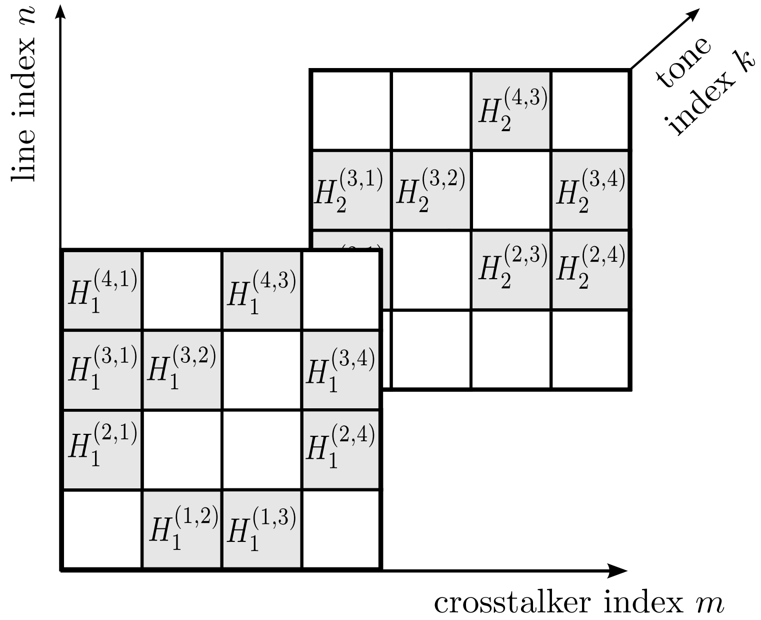


Figure 7.4: Example result of S-JTLS

For a given complexity limit C_{run}, S-JTLS can cancel

$$p_{max} = C_{run} \tag{7-17}$$

crosstalker-tone pairs in the binder. Depending on the data rate requirements, it distributes the available number of pairs which can be canceled to the different users, so that the total needed complexity of the binder is the sum of pairs p_n canceled for each user n.

The pseudo code for S-JTLS is given in Algorithm 4. It starts with calculating the capacity gain of each pair and sorting the pairs according to their gains, in order to estimate the performance gains achievable by canceling a certain crosstalker on a certain tone. Using again an approximation for $SINR_k^n$, the capacity gain is calculated by

$$g_n^{JTLS}(m,k) = \log_2\left(1 + \frac{1}{\Gamma} \cdot \frac{\left|H_k^{(n,n)}\right|^2 \cdot P_k^n}{\sigma^2}\right) - \log_2\left(1 + \frac{1}{\Gamma} \cdot \frac{\left|H_k^{(n,n)}\right|^2 \cdot P_k^n}{\left|H_k^{(n,m)}\right|^2 \cdot P_k^m + \sigma^2}\right) \tag{7-18}$$

with $n \neq m$. $g_n^{JTLS}(m,k)$ is the gain of user n when canceling crosstalker m on tone k. The sorted set of crosstalker-tone pairs is given by

$$\begin{gathered}\{d_n(1), \ldots, d_n((N-1) \cdot K_u)\} \\ \text{subject to: } g_n^{JTLS}(d_n(i)) \geq g_n^{JTLS}(d_n(i+1)),\ \forall i.\end{gathered} \tag{7-19}$$

Init: Calculate $g_n^{JTLS}(m,k)$, $\forall m,k$ for each user n.
  Sort $g_n^{JTLS}(m,k)$ in descending order for each n.
  $R_{n,JTLS}^{(C)} = R_n^{(I)}$, $j = 0$, $p_n = 0$, $\forall n$

while *(*$\sum\limits_n p_n < p_{max}$ *and* $R_n^{(T)} > R_{n,JTLS}^{(C)}$ *, for at least one user n)* **do**
  $j \leftarrow j+1$
  for $n = 1 \dots N$ **do**
    if $R_n^{(T)} > R_n^{(C)}(p_n)$ **then**
      Increase p_n: $p_n \leftarrow j \cdot \Delta$
      if $\sum\limits_n p_n \leq p_{max}$ *and* $p_n \leq (N-1) \cdot K_u$ **then**
        Calculate $R_{n,JTLS}^{(C)}(p_n)$
      end
      else
        $p_n \leftarrow (j-1) \cdot \Delta$
      end
    end
  end
end

if *(*$\sum\limits_n p_n < p_{max}$*)* **then**
  $\Delta p = p_{max} - \sum\limits_n p_n$.
  Sort $g_n^{JTLS}(m,k)$ of all uncanceled crosstalker-tone pairs of all users in descending order.
  Increase p_n of the users with the highest gains until $\Delta p = 0$.
end

Algorithm 4: Successive joint tone-line selection

$d_n(i)$ indicates user n's pair with the ith largest gain. In the following, the pairs p_n which should be considered in the cancellation procedure to fulfill the data rate requirements are successively selected. As long as there is computational complexity available, not all users achieved their desired data rates and cancellation is not performed fully, p_n is increased. In every iteration of the while-loop in Algorithm 4, p_n is raised to

$$p_n = j \cdot \Delta \tag{7-20}$$

by complexity factor Δ. The selection of canceled pairs is done based on the sorted gains

$g_n^{JTLS}(m,k)$.

After each complexity increase, fulfillment of data rate requirements is checked by calculating the current data rate $R_{n,JTLS}^{(C)}$ and comparing it to the target data rate $R_n^{(T)}$. For S-JTLS, the current data rate $R_{n,JTLS}^{(C)}$ is calculated by

$$R_{n,JTLS}^{(C)}(p_n) = f_S \cdot \sum_k \log_2 \left(1 + \frac{1}{\Gamma} \cdot \frac{\left| H_k^{(n,n)} \right|^2 \cdot P_k^n}{\sum\limits_{m \notin \mathbb{M}_{QoS,k}^{n,JTLS}} \left| H_k^{(n,m)} \right|^2 \cdot P_k^m + \sigma^2} \right). \tag{7-21}$$

When the algorithm exits the while-loop, the remaining complexity is given to the crosstalker-tone pairs with the highest gains, to maximally profit from the residual resources.

The outcome of S-JTLS is the selected set of crosstalkers $\mathbb{M}_{QoS,k}^n$ for each user n on each tone k. It is given by

$$\mathbb{M}_{QoS,k}^{n,JTLS} = \{m | (m,k) \in \{d_n(1), \ldots, d_n(p_n)\}\}. \tag{7-22}$$

7.1.4 Computational Complexity Analysis

The goal of the successive selection algorithms is to keep the needed run-time complexity for data rate fulfillment as low as possible. In this thesis, it is always limited to a certain amount C_{run} for the whole binder. The run-time complexity of crosstalk cancellation is the computational complexity spent for the additional signal processing during transmission. It is continuously needed, as crosstalk cancellation is performed for each single symbol. In order to save cost, it is desirable to minimize it while achieving an acceptable performance.

The DSL channel can vary slowly over time which can result in a change of crosstalker strength. The canceler and precoder have to be adapted to the channel changes and therefore the initialization complexity C_{init} also needs to be analyzed. It is the complexity needed to find the selected set of canceled crosstalkers $\mathbb{M}_{QoS,k}^n$ before transmission. To limit computation time and cost, it is desirable to keep this complexity low. At the same time it needs to be high enough to allow the selection procedure to find a set of crosstalkers which leads to the desired performance. Nevertheless, the procedure has to be repeated at a low frequency and computation time is not overly critical.

In this section, the initialization complexity for S-LS, S-TS and S-JTLS is analyzed. As explained in the last section, all proposed selection algorithms are successive, which makes

Table 7.1: Required initialization complexity

a) Sorting of disturbers and calculation of initial data rate			
	Multiplications	Sort operations	Capacities
S-LS	$N \cdot K_u \cdot (N-1)$	$N \cdot K_u \cdot (N-1) - sized$	$N \cdot K_u$
S-TS	$N \cdot K_u \cdot [(N-1)+5]$	$N \cdot K_u - sized$	$N \cdot K_u$
S-JTLS	$N \cdot 3K_u \cdot N$	$N \cdot K_u \cdot (N-1) - sized$	$N \cdot K_u$

b) Fulfillment of data rate targets	
	Capacities per user n and iteration (upper bound)
S-LS	K_u
S-TS	K_u
S-JTLS	K_u

c) Distribution of remaining complexity	
	Sort operations
S-LS	$[N \cdot K_u \cdot (N-1) - \sum_n c_n \cdot K_u] - sized$
S-TS	$[N \cdot K_u - \sum_n t_n] - sized$
S-JTLS	$[N \cdot K_u \cdot (N-1) - \sum_n p_n] - sized$

the initialization complexity dependent on the number of iterations the algorithm executes. Additionally, the initialization complexity spent within each iteration and the initialization complexity needed to distribute the remaining complexity after achievement of desired data rates inherit a dependency on chosen data rate targets and available run-time complexity.

Thus, only a rough estimation of required initialization complexity can be given in this section. Upper worst-case limits are provided and the number of iterations is kept as a parameter. Solely the most complex operations (multiplications, sort operations) are taken into account. The needed capacity calculations are obtained, which include one log operation, one multiplication and one division.

The three developed algorithms S-LS, S-TS and S-JTLS can generally be separated into three parts. In the first part at the beginning of the algorithms, the crosstalkers, tones or crosstalker-tone pairs are sorted based on their strength or gain and the initial data rate is calculated. Table 7.1 a) compares the needed initialization complexities all algorithms require for sorting of disturbers and calculation of the initial data rate. A worst case assumption is made as all users are taken into account. In practice, when there are users who are able to achieve their data rate requirements without any further signal processing, no sorting is necessary.

The second part of all algorithms is the while-loop. The number of executed operations depends on the number of iterations of the loop. The most complex operation in this part is the calculation of the current data rate. Table 7.1 b) shows the initialization complexity needed for this calculation for the three algorithms per iteration and user. It is only a worst-case estimation as generally, the current data rates will not need to be calculated for all users in each single iteration.

In the last part of S-LS, S-TS and S-JTLS, the remaining run-time complexity is spent on the uncanceled disturbers with the highest strength or gains. This requires sorting of all remaining crosstalkers, tones or crosstalker-tone pairs.

The needed initialization complexity given in Table 7.1 c) depends on the amount of crosstalkers, tones or crosstalker-tone pairs already considered in the cancellation procedure.

7.1.5 Performance

This section analyzes the performance of partial crosstalk cancellation using the newly developed successive selection algorithms S-LS, S-TS and S-JTLS. Analysis is done for a frequency-, as well as space-selective transmission scenario presented in Sec. 7.1.5.2. The algorithms are compared to a conventional selection algorithm, described in [CMA+03, CMG+04]. Comparison is done for different amounts of relative available run-time complexity C'_{run}. The relative run-time complexity is the relation of available run-time complexity C_{run} to the computational complexity needed for a full cancellation given by $N \cdot (N-1) \cdot K_u$. The simulation parameters are given in Sec. 7.1.5.1. Upstream results are shown in Sec. 7.1.5.3 and downstream results in Sec. 7.1.5.4.

7.1.5.1 Simulation Parameters

The simulation parameters are given in Table 7.2. No coding (i.e. a coding gain of 0 dB) is assumed. A symbol error probability of 10^{-7} is targeted. Perfect synchronization is presumed.

Table 7.2: Simulation parameters

Parameter	**Value**
Bandwidth B	17.664 MHz
Number of US DMT tones K_{US}	1147
Number of DS DMT tones K_{DS}	2885
Tone width Δf	4.3125 kHz
Coding gain γ_c	0 dB
Noise margin γ_m	6 dB
Target symbol error rate	$< 10^{-7}$
Transmit signal PSD	-60 dBm/Hz
Background noise PSD	-140 dBm/Hz
Band plan	998ADE17
Shannon gap $\Gamma(P_e)$	9.8 dB
Complexity parameter Δ	1147 in US
	2885 in DS

The channel model employed for the generation of the crosstalk channel coefficients is the beta model presented in Sec. 3.3. It considers the geometry of the binder.

According to the VDSL2 standard [Sta11] the tone spacing Δf is 4.3125 kHz. In practical systems the DMT symbol rate is $f_S = 4$ kHz. In these simulations the cyclic prefix is neglected, so that the achievable rates in practical systems are a little bit lower.

The Shannon gap $\Gamma(P_e)$ is given as 9.8 dB, together with a noise margin γ_m of 6 dB. According to Eq. 2-6 in Sec. 2.2.2, the SNR gap Γ results in 15.8 dB. Based on band plan FDD 998ADE17 [Sta11], FDD is used in the system. The frequency limits of the upstream and downstream bands are given in Table 7.3(a) and Table 7.3(b). All VDSL system standards

Table 7.3: Frequency bands

(a) Upstream frequency bands

Upstream band	**Frequency [kHz]**
US1	3750-5200
US2	8500-12000

(b) Downstream frequency bands

Downstream band	**Frequency [kHz]**
DS1	276-3750
DS2	5200-8500
DS3	12000-17664

assume that modems transmit with a spectral mask of -60 dBm/Hz. Therefore all modems transmit at this power level. Besides FEXT, the considered noise source is AWGN on each tone. Access to all line cards is assumed. The number of users is defined to be $N = 10$. The line diameter of each twisted pair is 0.4 mm (TP1).

7.1.5.2 Transmission Scenario and Definition of Data Rate Constraints

To analyze the performance of the different algorithms, a Distributed Line Length (DLL) scenario is considered which is schematically depicted in Figure 7.5. It presents a typical DSL transmission scenario as users generally have varying distances to the CO. Normally, a binder contains 20 to 100 twisted pairs. For a quantitative analysis, a lower number of twisted pairs can be chosen. The number of lines in the binder is defined to be $N = 10$, leading to 9 interfering lines for each user. The users are located at a distance from 0.3 km to 1.0 km to the CO with a linear increase of 0.0778 km from one line to the adjacent one. The transmission channels all users are facing are frequency- and space-selective both in upstream and downstream. Due to the varying lengths, the direct channel transfer function and the crosstalk channel transfer functions are different from line to line. All crosstalk channels are in addition influenced by the binder geometry which is considered in the applied crosstalk channel model (see Sec. 3.3). More information on the transmission properties on the different lines in the binder can be found in Appendix B.

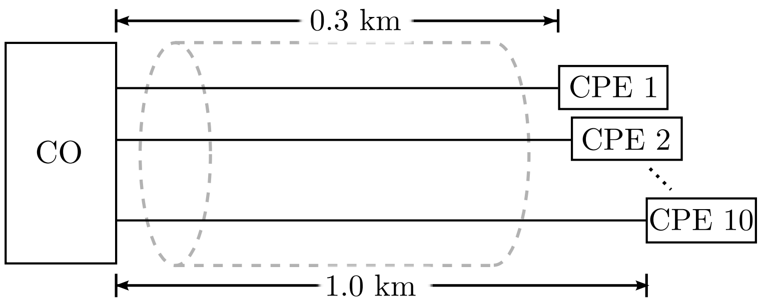


Figure 7.5: Distributed line length scenario

To analyze the scenario in terms of minimum and maximum possible data rates on the lines, the achievable upstream and downstream data rates for no and full crosstalk cancellation are given in Table 7.4. When no crosstalk is reduced in upstream, the lowest data rate is achieved by line 10. It is close to 0 Mbit/s. The highest data rate is achieved by user 1 with 55 Mbit/s. The data rates naturally decrease with line length due to larger attenuation. Nevertheless, line

Table 7.4: Upstream and downstream data rates in Mbit/s

US data rates										
line n	1	2	3	4	5	6	7	8	9	10
No XT canc.	55	43	28	28	15	1	3	4	4	0
Full XT canc.	73	66	59	52	44	36	28	21	14	10
DS data rates										
line n	1	2	3	4	5	6	7	8	9	10
No XT canc.	46	80	66	68	73	55	51	56	48	44
Full XT canc.	176	162	145	128	110	92	75	64	56	50

6 experiences a very low data rate as the binder geometry also influences the transmission channel. For full crosstalk cancellation, the data rates decrease from line 1 with 73 Mbit/s to line 10 with 10 Mbit/s. When all crosstalk is present in downstream, the highest achieved data rate is present on line 2 with 80 Mbit/s. The lowest achieved rate is 44 Mbit/s for line 10. Though being the shortest line, line 1 must have bad channel conditions due to interferences from other lines, as the achievable data rate is just 46 Mbit/s. For full crosstalk cancellation the data rates decrease from line 1 with 176 Mbit/s to line 10 with 50 Mbit/s.

The lines of the DLL scenario are divided into three groups of three, four and three lines, which have near (line 1-3), middle (line 4-7) and far (line 8-10) range because the achievable data rates decrease with line length when no crosstalk influences the transmission. To analyze and challenge the algorithms, data rate targets are chosen in a way that always some users initially achieve their target data rates and others need to come close to crosstalk-free transmission. In upstream the defined data rate targets for these three groups are $[55;25;5]$ Mbit/s as a high data rate target and $[40;20;5]$ Mbit/s as a low data rate target. In downstream they are $[140;75;45]$ Mbit/s and $[125;65;45]$ Mbit/s respectively. All data rate targets are shown together with the maximum and minimum possible data rates on the lines in Fig. 7.6 and Fig. 7.7. The high data rate targets are illustrated as dashed lines. The low data rate targets are drawn as solid lines.

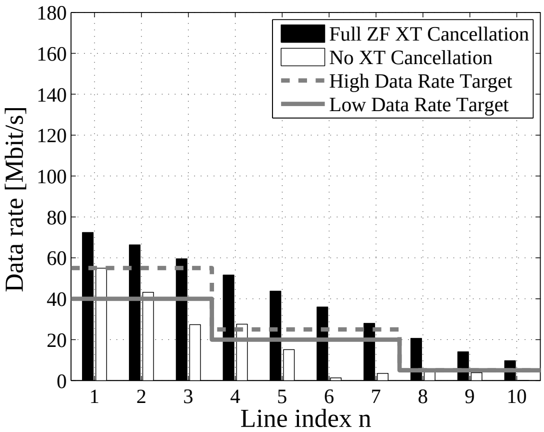


Figure 7.6: DLL scenario: US data rates

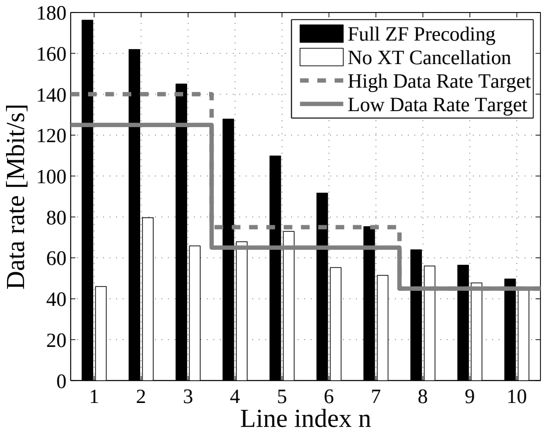


Figure 7.7: DLL scenario: DS data rates

7.1.5.3 Upstream Results

In this section, all successive selection algorithms are compared to each other and to a conventional selection algorithm in upstream. The comparison criterion between all algorithms will strictly be the percentage of users with achieved data rate target. The percentage of users with achieved data rate requirement is given as the number of users that have already fulfilled their target data rate over the total number of users N in percent. It is calculated and shown for all available run-time complexity values.

In upstream, partial crosstalk cancellation is performed on the receiver side. In the DLL scenario, all lines have different distances to the receivers. This leads to varying coupling lengths among lines and affects performance due to near-far effects. All users have different direct channels and the crosstalk channels differ due to binder geometry and varying coupling length.

7.1.5.3.1 Required Run-Time Complexity

Figure 7.8 and Fig. 7.9 show the percentage of users who achieved their target data rate depending on the available run-time complexity for high and low data rate targets. Partial crosstalk cancellation using S-LS, S-TS and S-JTLS is compared to application of conventional Simplified Joint Tone-Line Selection (JTLS), which is known to achieve near-optimum data rates in terms of capacity optimum of the binder. All four selection algorithms are able to fulfill the data rate targets, but all successive algorithms show better performance than the conventional algorithm. Figure 7.8 shows that for the high data rate

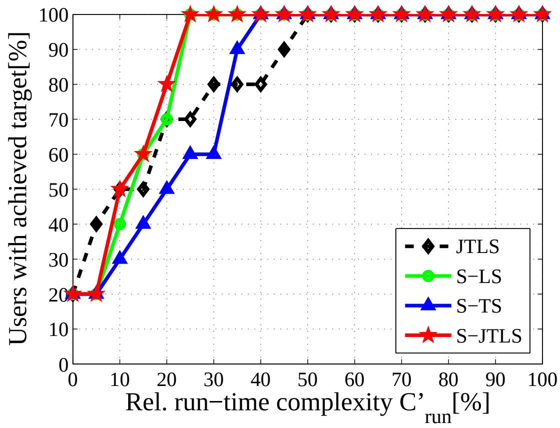


Figure 7.8: Comparison of needed complexity for high data rate target

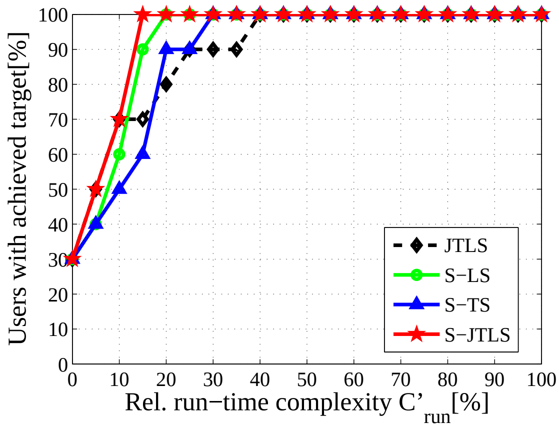


Figure 7.9: Comparison of needed complexity for low data rate target

target S-JTLS and S-LS only need 25% run-time complexity to allow all users to reach their data rate goals, where S-TS needs 40%. JTLS has worst performance with requiring 50% complexity for fulfillment of data rate requirements. One might recognize in Fig. 7.8, that for 5% complexity JTLS is able to satisfy 40% of the users and all successive algorithms only satisfy 20%. This can be explained by the ways the two algorithms distribute the available complexity. JTLS allocates each user the same amount of complexity. S-JTLS, S-LS and S-TS allocate larger portions of complexity to a single user and solely favour some lines when only 5% run-time complexity is available. In case of low data rate requirements (see Fig. 7.9), results are only slightly different. S-JTLS shows best performance with requiring 15% run-time complexity, where S-TS uses 30% and S-LS needs 20%. JTLS shows worst performance, as it needs 40% complexity to achieve all users' data rate targets.

To analyze how S-JTLS outperforms the other algorithms, Fig. 7.10 and Fig. 7.11 show the achievable rates on all lines for $C'_{run} = 25\%$ and $C'_{run} = 15\%$ respectively. When these run-time complexities are available S-JTLS allows all users to achieve their target data rates. For the high data rate target (see Fig. 7.10), S-JTLS and S-LS fulfill the data rate goals on all lines. It can be seen that S-JTLS nevertheless achieves higher data rates than S-LS for 25% run-time complexity, as it exceeds the target data rates more than S-LS does. S-TS and JTLS both do not reach the target data rates on all lines. JTLS exceeds the goal rates a lot for users 1, 4 and 8, but is far away for lines 3 and 7. In total, three lines are not able to fulfill their data rate goals for JTLS and a run-time complexity of 25%. S-TS does not much exceed the data rate goals for the lines that achieved their data rate requirement, but line 3, 6, 7 and 8

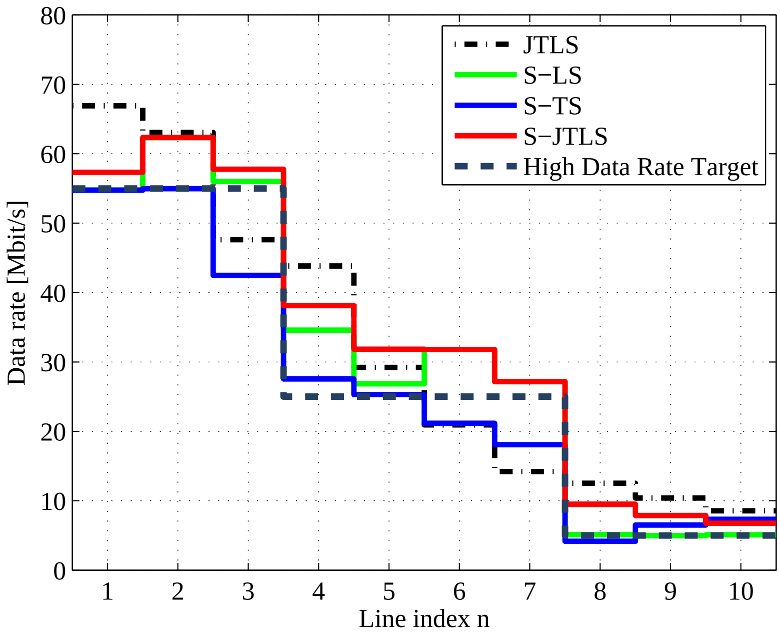


Figure 7.10: Upstream data rate comparison for C'_{run} = 25% and high data rate target

are below the goal rate for a run-time complexity of 25%. The data rate achieved by line 3 in Fig. 7.10 is much smaller than the data rate target. For the low data rate target (see Fig. 7.11), only S-JTLS is able to fulfill all data rate goals at a run-time complexity of 15%. For line 1 and 6 it massively exceeds the data rate target, which is because S-JTLS distributes the remaining complexity to the lines with the highest gains after the data rate targets are fulfilled. For S-LS only line 8 does not fulfill the data rate goal. Nevertheless, line 1 and 6 are well above the data rate requirement already. For S-TS, line 3, 6, 7 and 8 are not able to fulfill their data rate goals. As visible in Fig. 7.11, JTLS again is well above the desired data rates for many lines, but is far away from the desired data rate goals for line 6 and 7.

For both low and high data rate targets it can be seen that JTLS massively exceeds the desired data rates although some other lines are not even close to fulfilling their data rate requirements. The proposed successive algorithms are generally closer to the desired data rates for the observed run-time complexities. As a result, the successive selection algorithms lead to a lower required run-time complexity, compared to the conventional algorithm in comparison.

In the simulations of S-LS, S-TS and S-JTLS the number of loop iterations required in the algorithms to reach all data rate goals was counted. As explained in Sec. 7.1.4 that number

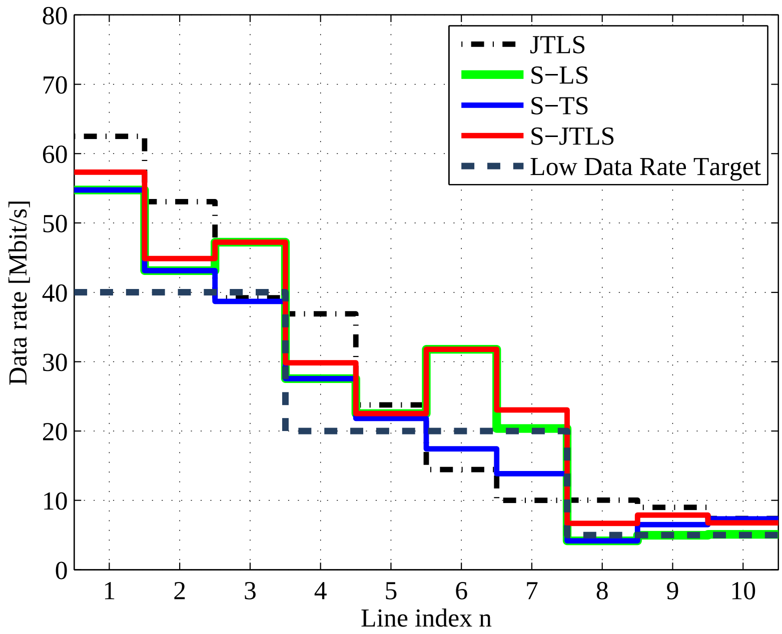
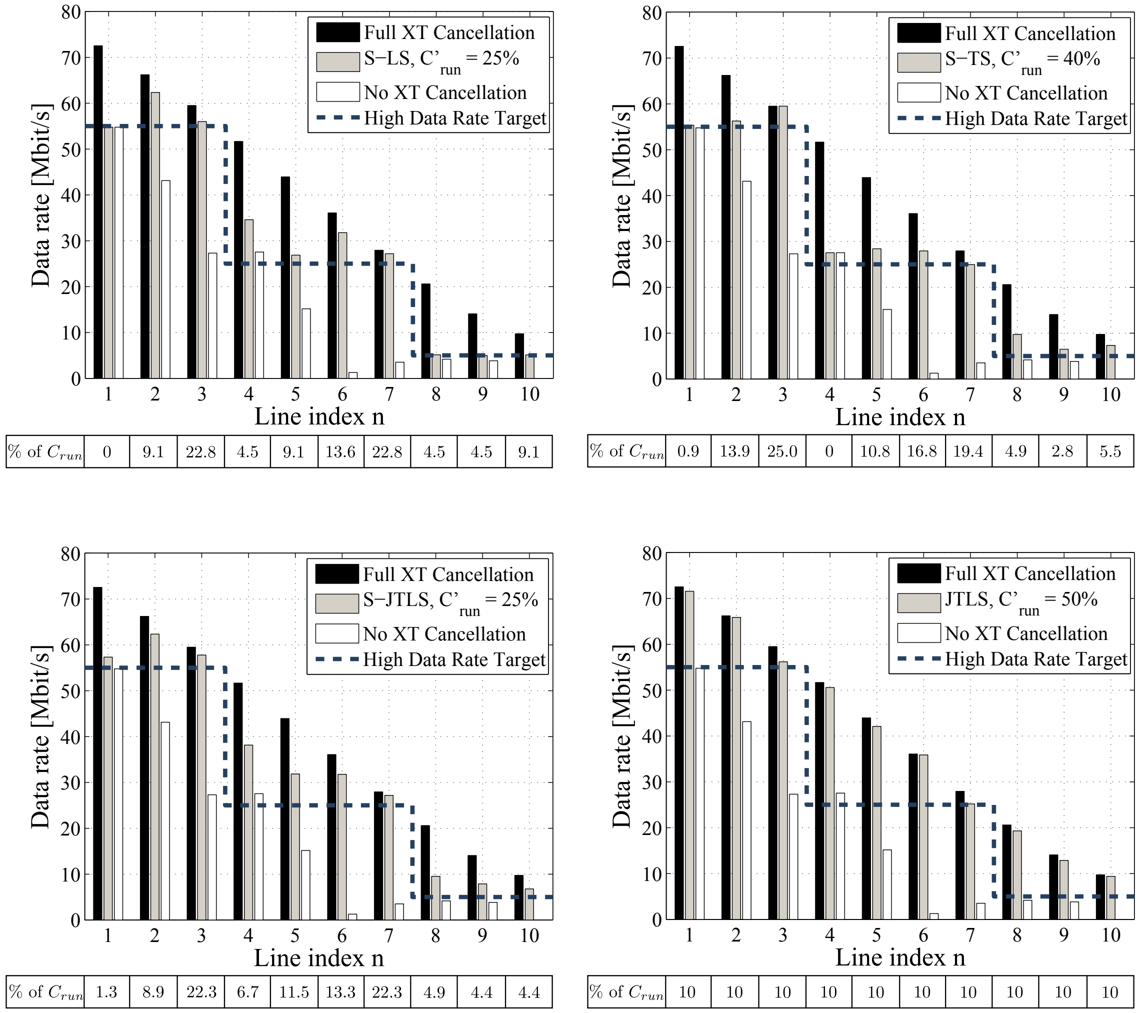


Figure 7.11: Upstream data rate comparison for C'_{run} = 15% and low data rate target

depends on the chosen scenario, the chosen target data rates and it also varies with the available run-time complexity. For both the high and the low data rate target, the maximum number of needed iterations was 10.

7.1.5.3.2 Run-Time Complexity per Line for Target Data Rate Fulfillment

In this paragraph, the complexity allocation to the different lines is analyzed when all lines achieved their high data rate targets. The minimum needed run-time complexity to satisfy all users is considered here as all successive algorithms distribute the resources which are not needed for target data rate fulfillment to the users with the highest gains. Nevertheless, there can be some complexity spent on the users with the highest gains, as the available complexity can be a bit higher than the one required to reach each user's data rate goal. In Fig. 7.12 the data rates of S-LS, S-TS, S-JTLS and JTLS are observed for C'_{run} equal to 25%, 40%, 25% and 50% respectively and compared to the achievable rates for no and full crosstalk cancellation. In addition, the tables show the percentage of absolute run-time complexity given to the individual lines. Figure 7.12 illustrates how the available complexity is distributed to the users for the chosen run-time complexities. In this case, all users achieved



% of C_{run} (S-LS)	0	9.1	22.8	4.5	9.1	13.6	22.8	4.5	4.5	9.1
% of C_{run} (S-TS)	0.9	13.9	25.0	0	10.8	16.8	19.4	4.9	2.8	5.5
% of C_{run} (S-JTLS)	1.3	8.9	22.3	6.7	11.5	13.3	22.3	4.9	4.4	4.4
% of C_{run} (JTLS)	10	10	10	10	10	10	10	10	10	10

Figure 7.12: Achieved data rates and percentage of run-time complexity distributed to the lines when all users achieved their target data rates

their data rate targets and exceeded them sometimes as shown by the bar plots. From the tables it can be seen, that all successive selection algorithms distributed the available complexity effectively, where JTLS gives the same amount of run-time complexity to all lines. User 4, who had sufficient data rate from the beginning is not allocated any run-time complexity for S-TS. S-LS and S-JTLS allocate 4.5% and 6.7% of run-time complexity to that user. They allocate remaining complexity to the users with the highest gains. User 1, which also initially reached his desired data rate, just gets no (S-LS) or only small amounts of run-time complexity below 1.5% (S-TS and S-JTLS). Line 3 and 7 together get nearly 50% of the total available complexity in all successive algorithms. Their target data rates are very close to the reachable data rate with full crosstalk cancellation, which indicates that many crosstalkers need to be canceled to fulfill the target data rates.

Fig. 7.13 exemplarily shows the receive PSDs for the upstream bands of user 3 and user 10. User 3 indeed has several crosstalkers of comparable strength, which requires a lot of them to be canceled to achieve near crosstalk-free performance. Line 10 also has a high data rate target and is strongly interfered. However, the right plot in Fig.7.13 illustrates that user 10 has a few very strong crosstalkers. Canceling them results in a big increase of data rate. This explains that user 10 requires only a modest amount of run-time complexity to achieve his data rate goal in comparison to line 3.

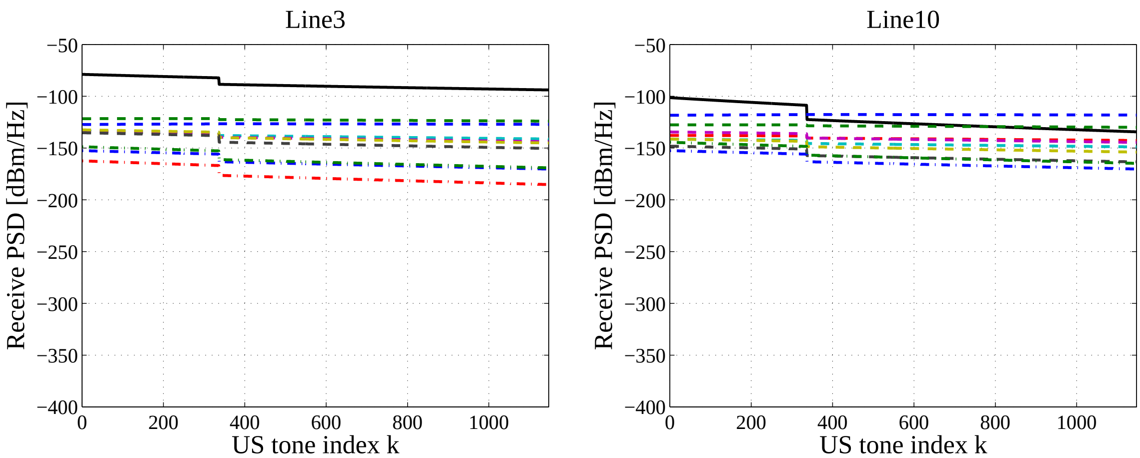


Figure 7.13: Direct channel (solid) and crosstalk channel (dashed) PSDs for line 3 and line 10

Table 7.5 compares the absolute amount of computational resources given to the users in the binder by the different selection algorithms when all achieved target fulfillment. For each selection algorithm it presents the number of crosstalker-tone pairs canceled per line. When the absolute amount of complexity resources given to a line is 1, it means that 1 crosstalker is canceled on 1 tone.

Table 7.5 again shows the advantage of the successive algorithms over the conventional selection algorithm. JTLS allocates the same amount of computational resources to each line. 5161 crosstalker-tone pairs are canceled for each line to fulfill all data rate targets. The successive algorithms are able to shift the available complexity to the users with high demands and spent the available run-time complexity only where it is needed. Since all successive algorithms stop the assignment of computational resources to the user who has already achieved a desired data rate, additional computational complexity can be shifted to users that require a more demanding data rate. The result is then that they provide a higher number of users with achieved data rate targets for a reduced overall computational complexity.

Table 7.5: Absolute run-time complexity spent per line to fulfill high data rate targets in US

line n	1	2	3	4	5	6	7	8	9	10
S-LS	0	2294	5735	1147	2294	3441	5735	1147	1147	2294
S-TS	360	5733	10323	0	4482	6921	8028	2016	1143	2286
S-JTLS	340	2294	5735	1729	2966	3441	5735	1273	1147	1147
JTLS	5161	5161	5161	5161	5161	5161	5161	5161	5161	5161

7.1.5.3.3 Achievable Data Rates

Fig. 7.14 shows the achieved data rates of all lines in the binder for a relative computational complexity from 0 to 100% for all successive algorithms and the high data rate targets. S-LS achieves the desired data rate at 25% run-time complexity, S-TS requires 40% and S-JTLS also 25% run-time complexity. In case of achieved data rate requirements for all users, all successive algorithms distribute the remaining complexity among all lines, depending on their gains. As can be seen in all figures, the available complexity was first used to fulfill the data rate requirements of all users as the data rates of lines which achieved their desired data rates remain stable until all received their data rate goals. For a complexity larger than 25% for S-LS, 40% for S-TS and 25% for S-JTLS all users achieved their target and the remaining complexity is given to the users which maximize the sum rate. This shows that the focus of the successive algorithms is on data rate fulfillment and not on rate maximization.

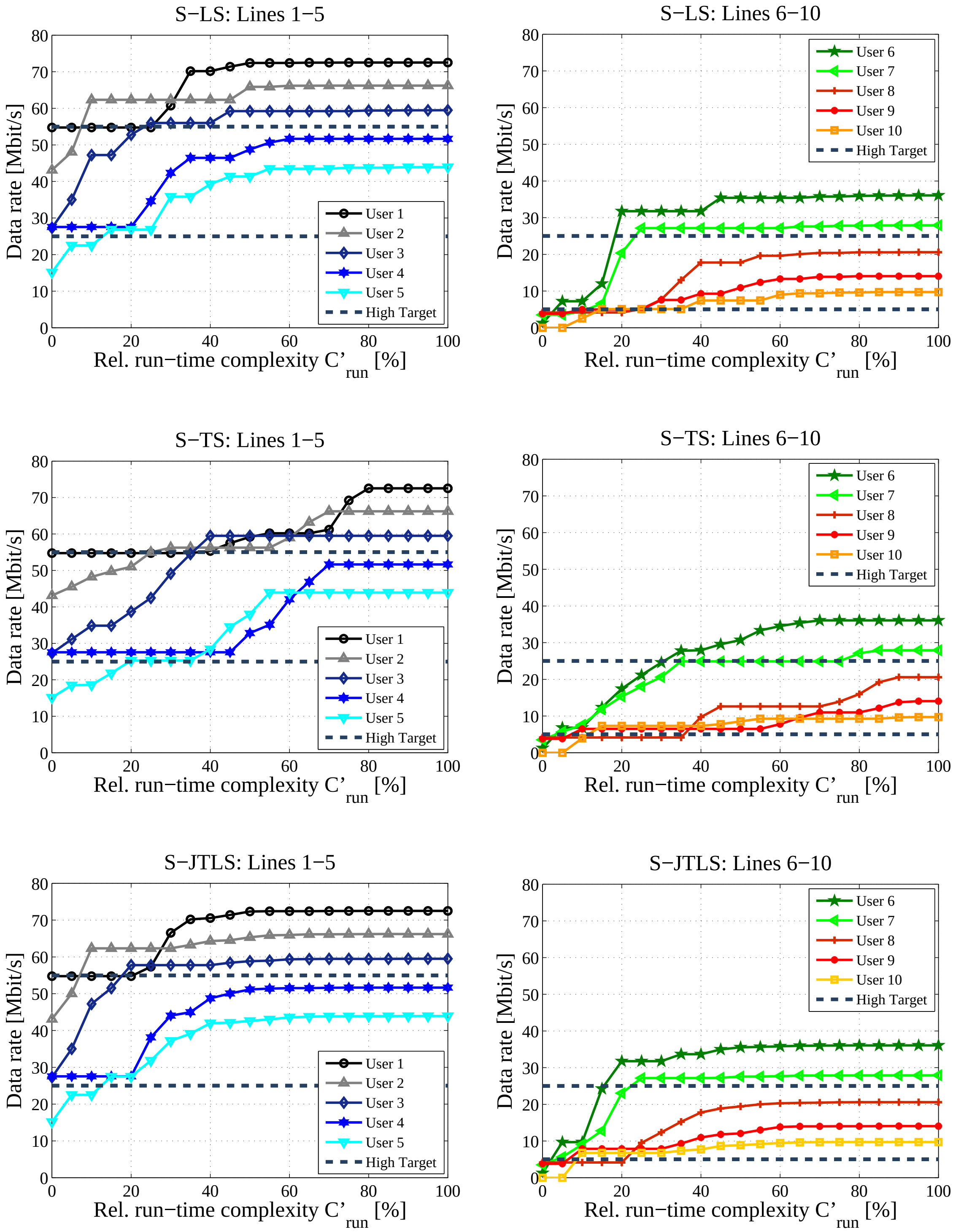


Figure 7.14: Achievable upstream data rates depending on available run-time complexity for S-LS, S-TS and S-JTLS

7.1.5.4 Downstream Results

In this section, the simulation results for all successive selection algorithms in downstream are presented. The performance of the algorithms is compared with each other and the performance of JTLS, a conventional selection algorithm. The comparison criterion will again be the percentage of users with achieved data rate target, which is the amount of users that have already achieved their desired data rates divided by the total number of users N. The number of satisfied users is calculated and shown for all available run-time complexity values.

Downstream partial precoding is performed on the transmitter side. Though the transmitters may have different distances to the receivers, the crosstalk coupling length is always the same and near-far effects do not occur. Consequently, different algorithm performance than in upstream is expected.

7.1.5.4.1 Required Run-Time Complexity

In Fig. 7.15 and Fig. 7.16 the percentage of users who achieved their data rate requirement depending on the available run-time complexity can be observed for high and low data rate targets. A comparison is made between conventional JTLS and the successive selection algorithms S-LS, S-TS and S-JTLS. All four algorithms are able to fulfill the desired data

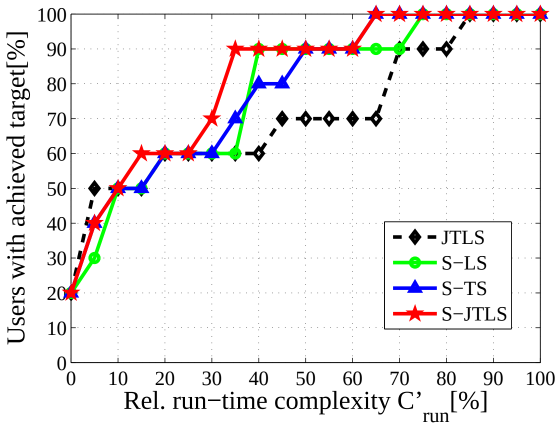


Figure 7.15: Comparison of needed complexity for high data rate target

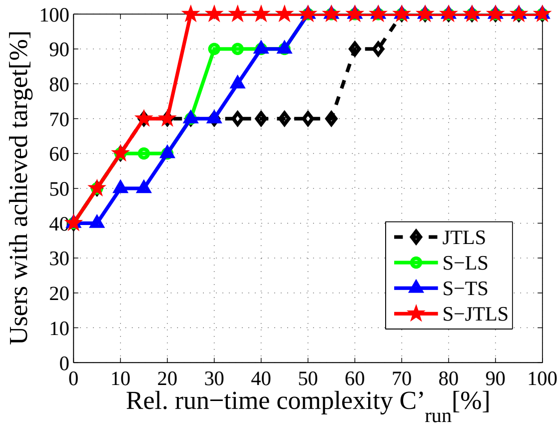


Figure 7.16: Comparison of needed complexity for low data rate target

rate targets, but the successive algorithms achieve better performance than the conventional algorithm. For the high data rate target (see Fig. 7.15), S-TS and S-JTLS require 65% of run-time complexity to fullfill all data rate requirements, where S-LS needs 75% and JTLS even requires 85%. Like in upstream, for 5% run-time complexity JTLS is able to satisfy more users than the successive algorithms. It satisfies 50% of the users, where S-JTLS and S-TS satisfy 40% and only 30% reach their data rate goals with S-LS. The reason for that can be found in the way, the algorithms distribute the available complexity resources. JTLS allocates each user the same amount of complexity. S-LS, S-TS and S-JTLS allocate defined portions of complexity to the users in every loop iteration. When only 5% run-time complexity is available not every line is allocated one of these portions.

In Fig. 7.16 for low data rate demands, the results look different from the results for high data rate demands. S-JTLS performs much better. It shows best performance with requiring only 25% of run-time complexity. S-LS and S-TS require double the run-time complexity with 50%. JTLS even needs 70% complexity to achieve all target data rates. In summary, it can be stated that for both low and high data rate targets the proposed successive selection algorithms require a lower run-time complexity than the conventional algorithm to fufill their data rate demands.

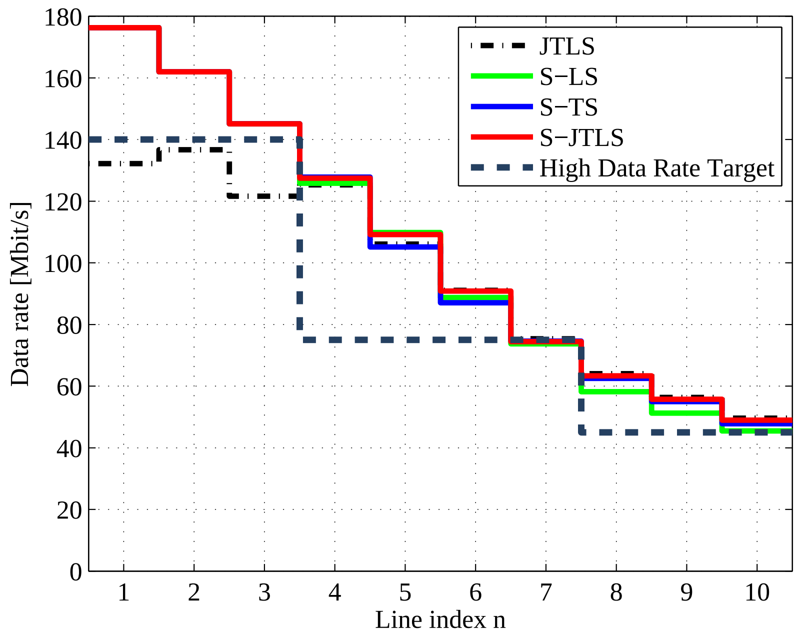


Figure 7.17: Downstream data rate comparison for C'_{run} = 65% and high data rate target

Fig. 7.17 shows the achievable data rates on each line for the high data rate targets and for a run-time complexity of 65% at which S-JTLS is able to satisfy all users. S-TS and S-JTLS both allow all users to fulfill their data rate goals. It is notable that they massively exceed the desired data rates. A lot of complexity must be spent on the lines achieving the highest gains to maximize the binder capacity at the end of the algorithms. For S-LS the situation is comparable for all lines except line 7. For $C'_{run} = 65\%$, it slightly misses the target data rate for that user. Non-successive JTLS delivers data rates much higher than the desired data rate goals except for line 1, 2 and 3. The users belonging to these lines are not able to reach their data rate goals.

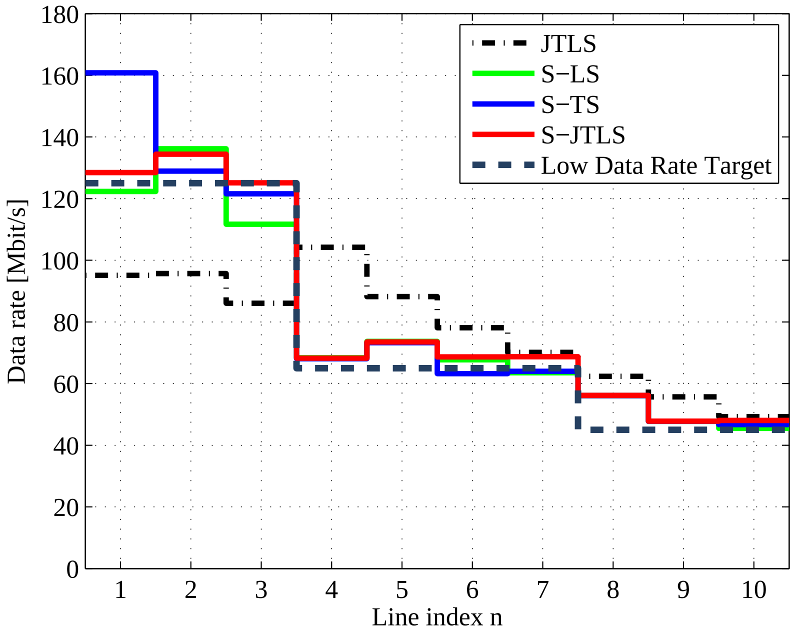


Figure 7.18: Downstream data rate comparison for C'_{run} = 25% and low data rate target

Figure 7.18 shows the achievable data rates for low data rate demands. The chosen run-time complexity is 25%. At this run-time complexity only S-JTLS is able to fulfill all data rate targets. S-LS does not reach the rate goals of users 1, 3 and 7. The achieved data rates of users 2, 5 and 8 are well above the desired data rate. For S-TS, the data rate of line 1 is 35 Mbit/s above the target. On the other hand users 3, 6 and 7 do not achieve their rate goals. JTLS shows worst performance with being far from the desired data rates for lines 1, 2 and 3, but high above the data rate goal for all other lines.

It can be stated, that for both high and low data rate target the successive selection algorithms outperform the conventional solution in terms of required run-time complexity to achieve the data rate requirements. For the high data rate demand at a complexity of 65%, they deliver data rates well above the desired data rate, comparable to the behaviour of JTLS. However, all lines equally are getting close to or higher than their goal data rate. JTLS leaves some users behind in achieving their goals. For the low data rate targets and 25% run-time complexity, the successive solutions generally deliver results closer to the data rate goals. The total overshoot in data rate is much lower than for JTLS.

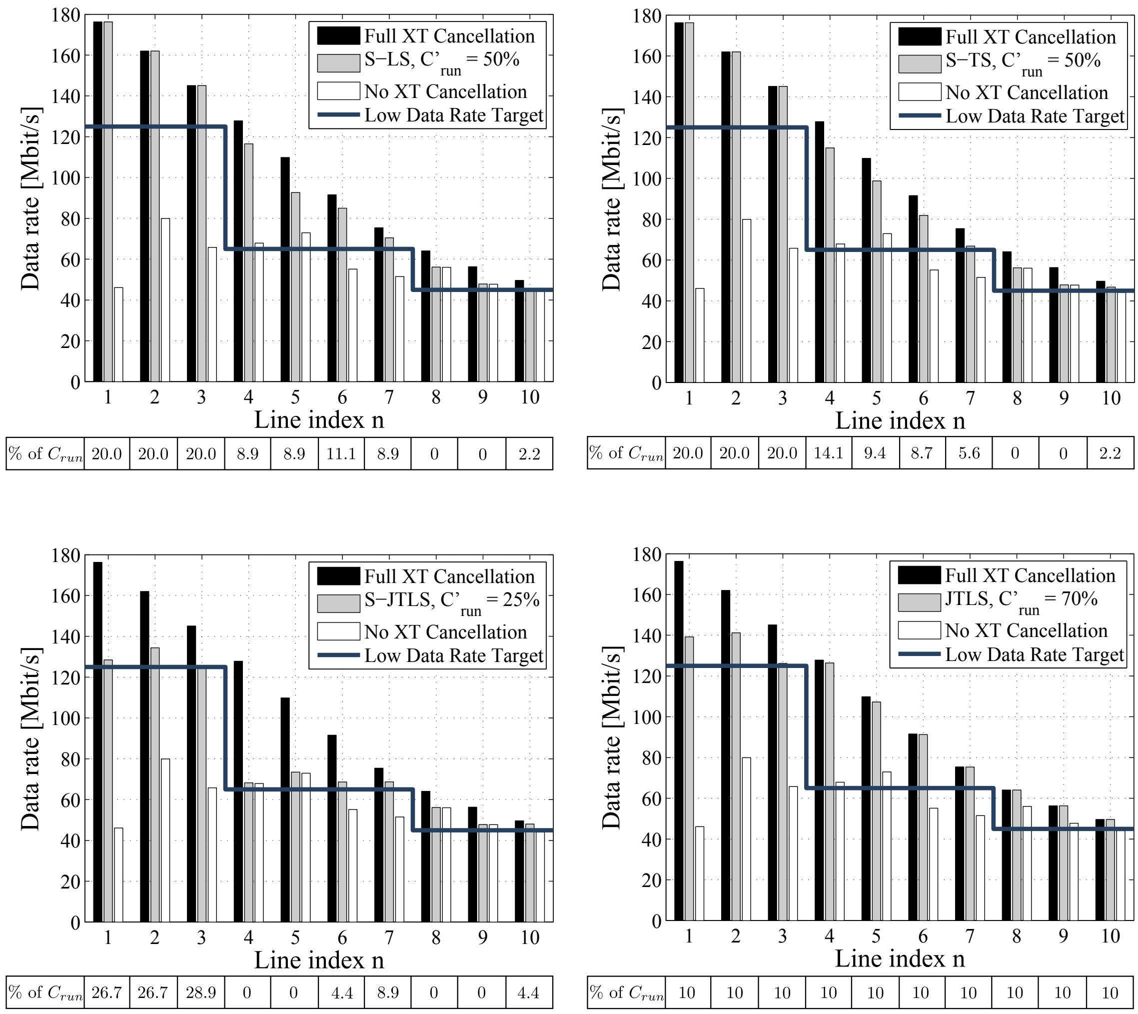
In downstream, the number of iterations required by the successive algorithms S-LS, S-TS and S-JTLS to exit the loop for target fulfillment was counted. For the chosen scenario and both the high and the low data rate target the maximum number of needed iterations was 10.

7.1.5.4.2 Run-time Complexity Per Line for Target Data Rate Fulfillment

This section analyzes the percentage of run-time complexity which is spent on the lines in the binder when all users achieved their high or low data rate targets. The minimum needed run-time complexity is observed, as all successive algorithms distribute the resources which are not needed to reach the data rate requirements to the users with the highest gains. Therefore, the performance of S-LS, S-TS, S-JTLS and JTLS is observed for 50%, 50%, 25% and 70% run-time complexity for the low data rate targets and at 75%, 65%, 65% and 85% run-time complexity of the high data rate targets.

Figure 7.19 shows how the available complexity is distributed to the users in the binder for low data rate demand. In the tables of Fig. 7.19 it can be seen that all successive selection algorithms distribute the available run-time complexity in an effective way compared to JTLS. Joint tone-line selection distributes the same amount of complexity to all lines. For the successive solutions the amount of run-time complexity given to the longer lines is lower. The short lines are given a higher percentage of run-time complexity. For S-JTLS the complexities are well-distributed amongst the lines in the binder as all lines hit their data rate goals quite closely. No unnecessary run-time complexity is spent. S-LS and S-TS give a lot of complexity to the short lines as these lines all achieve their maximum rates.

Table 7.6 shows the absolute run-time complexity available for the users in the binder to allow them to achieve their desired data rates. The absolute run-time complexity is the number of crosstalker-tone pairs available for cancellation. For full cancellation the interferences of $25965 = 9\ (\text{crosstalkers}) \cdot 2885\ (\text{tones})$ crosstalker-tone pairs would be eliminated. S-JTLS shows the best performance as it needs the lowest amount of absolute run-time complexity. For S-LS and S-TS it can be seen that they fully cancel the first three lines. A lot more



% of C_{run}	20.0	20.0	20.0	8.9	8.9	11.1	8.9	0	0	2.2

% of C_{run}	20.0	20.0	20.0	14.1	9.4	8.7	5.6	0	0	2.2

% of C_{run}	26.7	26.7	28.9	0	0	4.4	8.9	0	0	4.4

% of C_{run}	10	10	10	10	10	10	10	10	10	10

Figure 7.19: Percentage of run-time complexity distributed to the lines in DS when all users achieved their desired data rate for low data rate targets

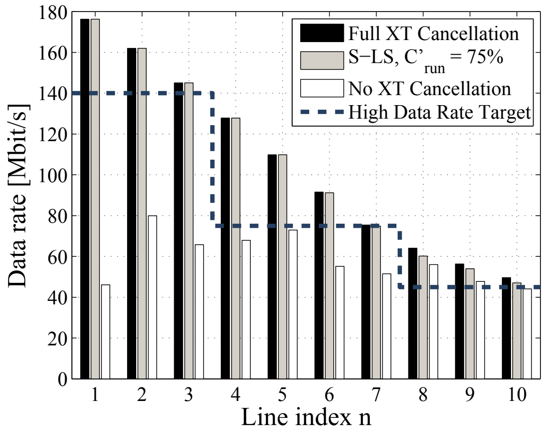
Table 7.6: Absolute run-time complexity spent per line to fulfill low data rate targets in DS

line n	1	2	3	4	5	6	7	8	9	10
S-LS	25965	25965	25965	11540	11540	14425	11540	0	0	2885
S-TS	25965	25965	25965	18261	12258	11286	7245	0	0	2880
S-JTLS	17310	17310	18752	0	0	2885	5770	0	0	2885
JTLS	18175	18175	18175	18175	18175	18175	18175	18175	18175	18175

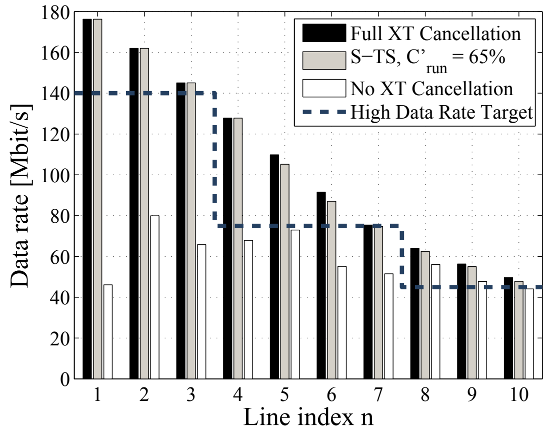
complexity than actually needed for data rate fulfillment is spent on these lines. The same result can be observed for line 4 and 5. For the short lines, S-LS, S-TS and S-JTLS show comparable performance with spending no or only low amounts of complexity to those lines. Joint tone-line selection equally distributes the available complexity amongst the lines in the binder. A lot more complexity than needed is spent to users 4 and 5 and to all short lines.

Figure 7.20 shows the percentage of complexity spent on the users in the binder for high data rate demands. As can be seen in the tables of Fig. 7.20, all successive selection algorithms effectively distribute the available run-time complexity to the lines in the binder for data rate fulfillment. In contrary, JTLS gives the same amount of run-time-complexity to every line. For the successive solutions, the amount of run-time complexity given to the longer lines is lower. The amount of run-time complexity given to the shorter lines is higher. Again especially the short lines come close to the maximum rates when all data rate targets are achieved as can be seen in the bar plots of Fig. 7.20.

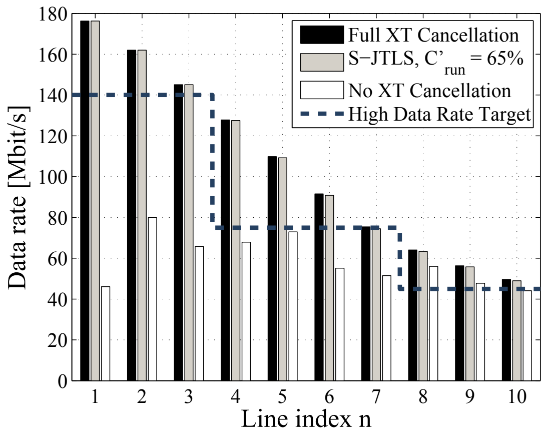
Table 7.7 shows the number of crosstalker-tone pairs selected for cancellation by the four selection algorithms to allow all users to achieve their desired data rates. All successive algorithms give lower amounts of resources to the long lines compared to JTLS. However, more complexity than actually needed for target data rate fulfillment is spent on the shortest three lines as they are fully canceled. This indicates that all successive algorithms already started distributing complexity resources to the users with the highest gains.



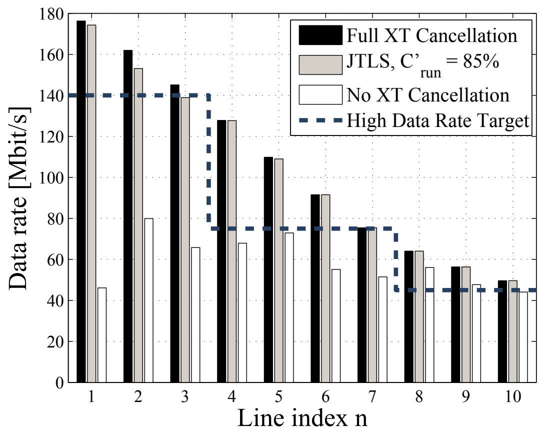
% of C_{run}	13.4	13.4	13.4	12.0	13.4	10.4	10.4	4.5	6.0	3.0



% of C_{run}	15.4	15.4	15.4	15.4	10.9	8.6	8.4	4.1	4.0	2.4



% of C_{run}	15.4	15.4	15.4	12.1	13.7	9.2	7.0	5.0	3.9	2.9



% of C_{run}	10	10	10	10	10	10	10	10	10	10

Figure 7.20: Percentage of run-time complexity distributed to the lines in DS when all users achieved their desired data rates for high data rate targets

Table 7.7: Absolute run-time complexity spent per line to fulfill high data rate target in DS

line n	1	2	3	4	5	6	7	8	9	10
S-LS	25965	25965	25965	23080	25965	20195	20195	8655	11540	5770
S-TS	25965	25965	25965	25965	18369	14535	14139	6903	6786	4176
S-JTLS	25965	25965	25965	20442	23090	15521	11843	8390	6673	4918
JTLS	22070	22070	22070	22070	22070	22070	22070	22070	22070	22070

7.1.5.4.3 Achievable Data Rates

Figures 7.21 and 7.22 show the achieved data rates of all lines in the binder for a relative computational complexity from 0 to 100%. The performance of all successive algorithms is analyzed. Figure 7.21 evaluates the achievable data rates for the low data rate demands. S-LS achieves all desired data rates at 50% run-time complexity, but already at 30% run-time complexity all lines except one reached their target. Line 7 is slightly below its desired rate. Although line 7 is still below its target, the algorithm assumes that all targets are fulfilled and starts distributing complexity again to the lines which already fulfilled their target. This is based upon the calculation of the current data rate in the algorithm. It is based on an estimation of the SINR. The complexity is given to the users with the highest gains, which explains that the short lines are favored. S-TS also requires 50% of run-time complexity to reach all target data rates. It shows a comparable behavior to S-LS as it also starts distributing resources to lines with achieved target at $C'_{run} = 30\%$. At that complexity lines 3, 6 and 7 did not achieve their target rates, but are really close to it. S-JTLS also requires 65% run-time complexity and shows comparable behavior to S-TS. S-JTLS does not suffer from inaccuracies of current rate calculations. It only distributes resources to lines with sufficient data rate after all lines achieved their targets.

Fig. 7.22 shows the achieved data rates of all lines in the binder for a high data rate demand.

S-LS achieves the desired data rate at 75% run-time complexity, but already at 40% run-time complexity all lines except one reached their target. Line 7 is only slightly below its desired rate. What can especially be seen in the plot for the short lines is that the algorithm assumes that all targets are fulfilled for $C'_{run} = 40\%$ as the data rates increase for lines 1, 2 and 4 which already reached their data rate goal. Run-time complexity is spent on the lines with the highest gains due to non-perfect estimation of the current data rate. S-TS requires 65% of run-time complexity, but also at $C'_{run} = 40\%$ the algorithm starts giving resources to the lines with the highest gains, as the algorithm assumes all users fulfilled their target. For 40% of complexity all except line 3 and 7 achieved their data rate goal. The data rates of line 3 and line 7 are close to the desired rate. Obviously, they do not have high gains, as line 3 gets allocated the missing resources for target fulfillment at 50% run-time complexity and line 7 at 65% run-time complexity. S-JTLS also requires 65% run-time complexity and shows comparable behavior to S-TS. But it performs slightly better as for $C'_{run} = 35\%$ all users except user 7 reached their desired rates. S-JTLS assumes that all lines achieved their target at that run-time complexity as it starts to give resources to the lines which already reached their data rate goals.

It can be summarized, that for both low and high data rate targets the available complexity was first used to approach the data rate goals of all users. The data rates of lines with achieved target remain stable until all users received their data rate goals or are quite close to it. Nevertheless, when the accuracy of the estimation of the current data rate is reached the algorithms can distribute complexity resources to lines with fulfilled targets also if not all lines fully achieved their rate goals.

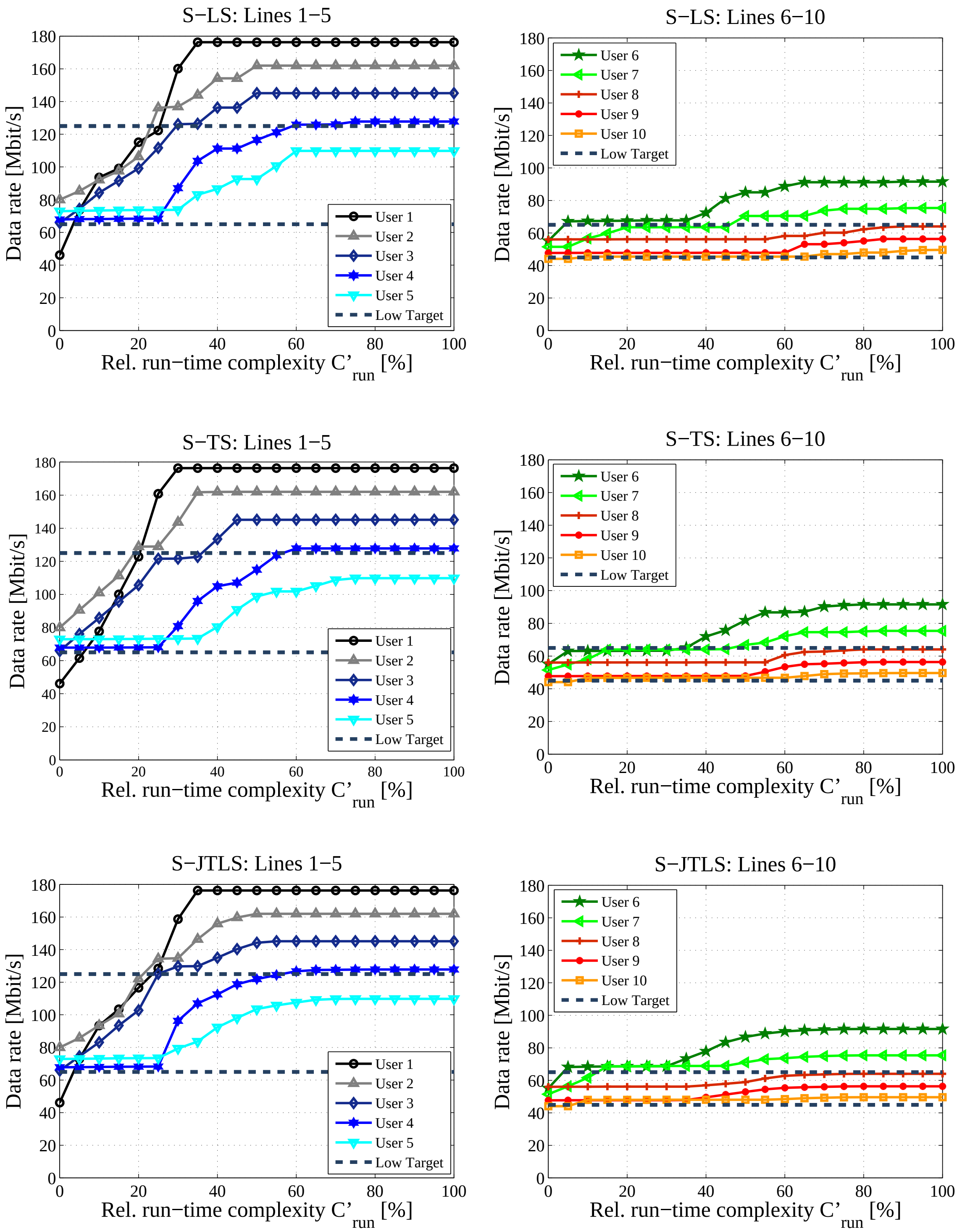


Figure 7.21: Achievable downstream data rates depending on available run-time complexity for S-LS, S-TS and S-JTLS and low data rate targets

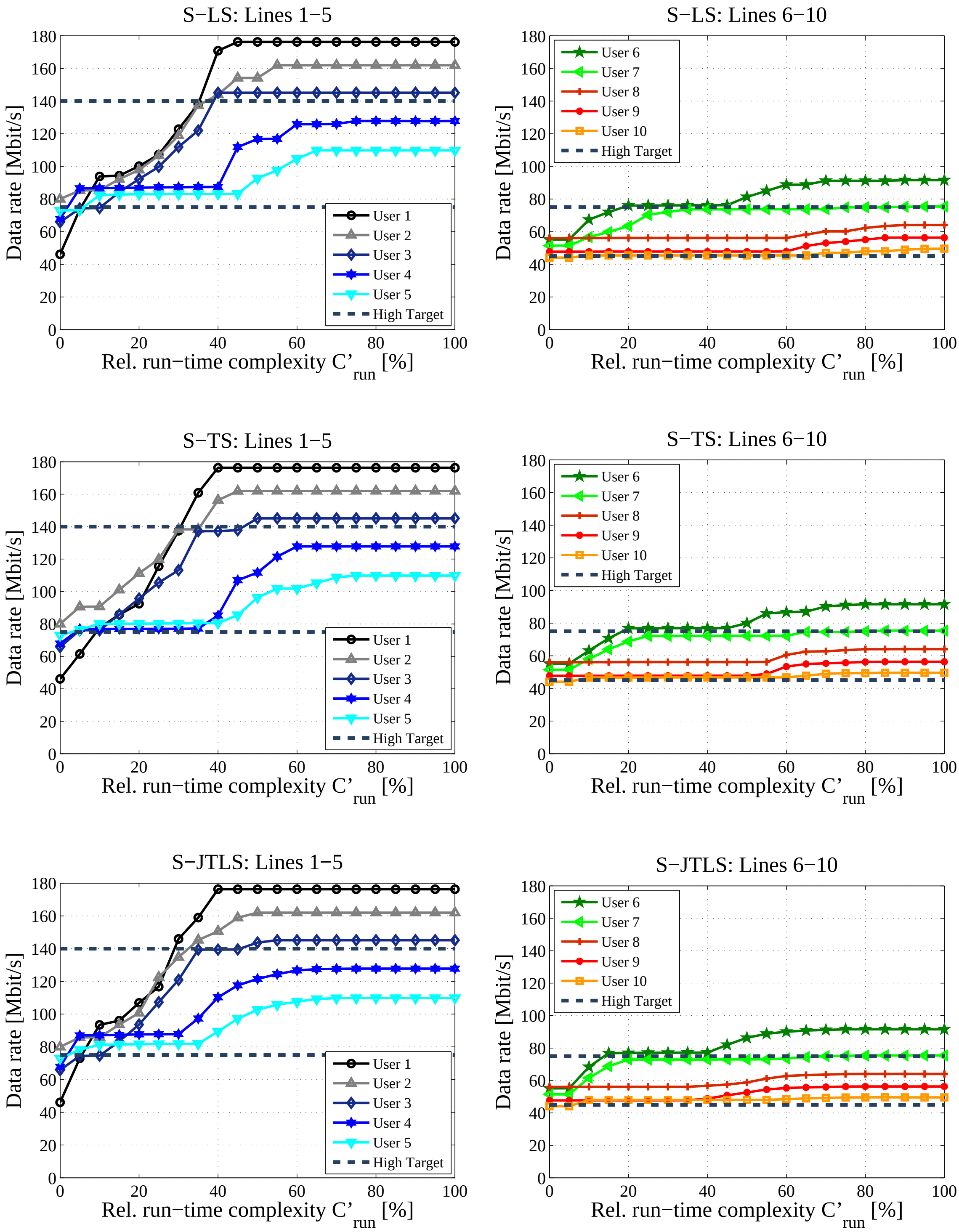


Figure 7.22: Achievable downstream data rates depending on available run-time complexity for S-LS, S-TS and S-JTLS and high data rate targets

7.2 Joint Partial Crosstalk Cancellation and Spectrum Management

In the last section partial cancellation procedures were presented, which are able to support data rate requirements of all users in a binder, for a limited computational complexity available for crosstalk cancellation. Successive selection algorithms decide on the interferers and tones which should be considered in the cancellation or precoding procedure to allow target data rate fulfillment to all users in the binder.

In Chapter 5, spectrum management techniques were analyzed, which are able to increase binder performance without any additional burden on the run-time complexity of DSL systems. Especially for transmission scenarios, in which the transmitters have differing distances to the receiver, they are able to increase performance.

Downstream crosstalk precoding is already capable of shaping the spectra of the transmit signals so no further improvement is expected there. In contrary, upstream crosstalk cancellation is performed on the receiver side and does not shape the spectra at all. Consequently, partial crosstalk cancellation is combined with spectrum management for upstream transmission. It has been shown in [BEP04, MMS09], that jointly combining both techniques outperforms the independent combination. However, none of these proposals considered any data rate constraints.

In this section, a joint spectrum management and partial cancellation approach is proposed which aims at minimizing the needed run-time computational complexity for DSL systems with data rate constraints. This proposal was also published by the author in [Dün12]. In Sec. 7.2.1 a selection algorithm is presented, which combines spectrum management with partial crosstalk cancellation. The basis of the selection algorithm is successive joint tone-line selection as presented in Sec. 7.1.3.3, due to its good performance in various target rate situations. For spectrum management, iterative waterfilling is considered as it works autonomous, is simple, has a low initialization complexity and does not need a reference line. Iterative waterfilling was already presented in Sec. 5.2.1.

7.2.1 Joint Successive Tone-Line Selection and Iterative Waterfilling

The joint approach of a selection algorithm, named Joint Successive Tone-Line Selection and Iterative Waterfilling (Joint S-TLS+IWF), combines the spectrum management solution iterative waterfilling with successive joint tone-line selection. As shown in Algorithm 5, it consists of two loops which both are left in case of convergence. In the outer loop S-JTLS is performed until the complexities remain stable. Here, the crosstalkers which need to be

Init: Set $P_k^n, \forall n$ to a flat PSD spectrum.
while *(No complexity convergence)* **do**
  Perform S-JTLS with latest P_k^n.
  while *(No spectra convergence)* **do**
    for $n = 1 \ldots N$ **do**
      Calculate the noise spectrum $\sigma^2_{n,total}$.
      Get P_k^n, $\forall k$ by waterfilling.
    end
  end
end

Algorithm 5: Joint S-TLS+IWF for data rate fulfillment

considered in the cancellation procedure to fulfill data rate targets are obtained. In the first iteration of this loop, flat transmit power density spectra are assumed for all users. With a fixed number and configuration of canceled crosstalker-tone pairs per line obtained by S-JTLS, the inner loop is entered. It works on spectra optimization with iterative waterfilling and is left when the spectra converge because the desired accuracy is reached. Starting with a flat spectrum, the first user virtually cancels the p_n designated crosstalker-tone pairs. The total noise contribution $\sigma^2_{n,total}$ of user n is calculated on each tone k. It contains additive noise and the remaining interferences from all other users and is calculated by

$$\sigma^2_{n,total}(k) = \sum_{\text{m not canceled}, n \neq m} |H_k^{(n,m)}|^2 \cdot P_k^m + \sigma^2 \tag{7-23}$$

where P_k^m is the transmit power of user m on tone k. Based on the noise contribution $\sigma^2_{n,total}$ on each tone k the first user water fills as explained in Sec. 5.2.1 to maximize his achievable data rate. The resulting transmit power of user n on all tones is given by

$$P_k^n = \frac{1}{K_{US}^*} \cdot \left[P_{max}^n + \sum_{k=1}^{K_{US}^*} \frac{\Gamma}{\mathrm{gn}_k^n} \right] - \frac{\Gamma}{\mathrm{gn}_k^n}, \tag{7-24}$$

with gain factor $\mathrm{gn}_k^n = |H_k^{(n,n)}|^2 / \sigma^2_{n,total}(k)$ and the number of used tones K_{US}^* which achieve $P_k^n > 0$. Due to the new power allocation of users n, $\sigma^2_{n,total}$ changed for all other users in the binder. This is considered in the repetition of the loop for each user. The algorithm does not exit the inner loop until the spectra converge and the desired accuracy is reached. In the next step, the outer loop is repeated. It optimizes the distribution of the complexities spent on partial cancellation on the different lines based on their target data rate. It iterates until the number of canceled pairs per line converges.

7.2.2 Computational Complexity Analysis

In Sec. 7.1.4 the computational complexity of S-JTLS was analyzed, where it was differed between initialization complexity C_{init} and run-time complexity C_{run}.
Joint successive tone-line selection and iterative waterfilling does not put any additional burden on the run-time complexity C_{run} but leads to an increase of the initialization complexity C_{init} as the optimum spectra have to be obtained. The algorithm contains two loops, where the complexity of the inner loop is given by the complexity of iterative waterfilling. Iterative waterfilling is well known to have approximately linear complexity in lines and tones [YGC02]. Normally, it converges in less than 10 iterations.
The number of operations needed to finish the algorithm depends on the iterations of the outer loop. Each loop contains operations to perform IWF and S-JTLS.

7.2.3 Performance

In this section the performance results of joint partial crosstalk cancellation and spectrum management are shown. Analysis is done for scenarios with varying line lengths as spectrum management is well known for increasing the performance in scenarios where long lines are heavily disturbed by the crosstalk of shorter lines. Only upstream transmission is considered as downstream precoding already shapes the transmit spectra. The Joint S-TLS+IWF selection algorithm is compared to S-JTLS without spectrum management presented in Sec. 7.1.3 and to sequential processing of IWF and S-JTLS. Simulation parameters are given in Sec. 7.2.3.1. Results are shown in Sec. 7.2.3.3.

7.2.3.1 Simulation Parameters

The simulation parameters are shown in Table 7.8. A target symbol error rate of 10^{-7} is assumed, so that the Shannon gap $\Gamma(P_e)$ can be calculated as 9.8 dB. No coding is used, hence the coding gain is 0 dB. An FDD system is considered. Perfect synchronization is presumed and the cyclic prefix is neglected. The channel model employed for the generation of the crosstalk channel coefficients is the beta model presented in Sec. 3.3. The maximum available power per user is $P^n_{max} = 6.94\,\text{dBm}, \forall n$. No spectral mask is applied and the granularity of bit loading is infinite. For all algorithms in comparison, which do not apply spectrum

Table 7.8: Simulation parameters

Parameter	Value
Bandwidth B	17.664 MHz
Number of US DMT tones K_{US}	1147
Tone width Δf	4.3125 kHz
Coding gain γ_c	0 dB
Noise margin γ_m	6 dB
Target symbol error rate	$< 10^{-7}$
Background noise PSD	-140 dBm/Hz
Band plan	998ADE17
Shannon gap $\Gamma(P_e)$	9.8 dB
Complexity parameter Δ	1147

management, the assumed spectrum is flat with -60 dBm/Hz. Only upstream transmission is considered with upstream band frequencies given in Table 7.9. Upstream band 0 is not used.

7.2.3.2 Transmission Scenarios and Definition of Data Rate Constraints

As spectrum management is well-known to show good performance in scenarios where near-far effects are present, the same DLL scenario as presented in Sec. 7.1.5 is considered. In the chosen scenario, the number of lines in the binder is defined to be $N = 10$, where the CPEs distance from the CO ranges from 0.3 km to 1.0 km with a linear increase of 0.0778 km from one line to the other. It is again schematically depicted in Fig. 7.23.

Like in the earlier section, the users are divided into three groups of 3/4/3 users: with high (line 1-3), with middle (line 4-7) and with low (line 8-10) demand. In these evaluations target data rates of 55, 25 and 5 Mbit/s are considered for the three groups. The data rates

Table 7.9: Upstream frequency bands

Upstream band	Frequency [kHz]
US1	3750-5200
US2	8500-12000

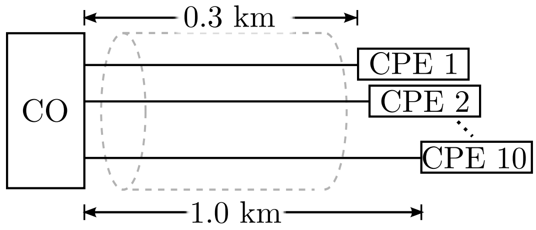


Figure 7.23: DLL scenario

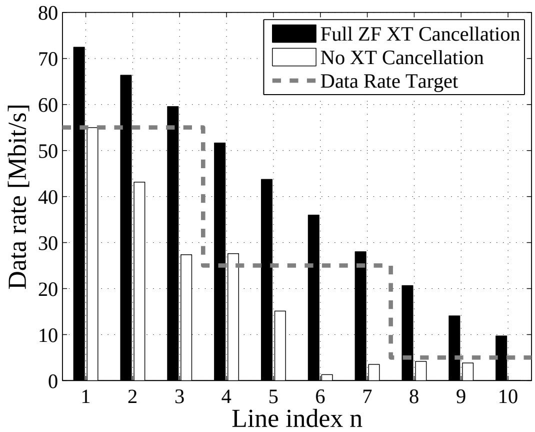


Figure 7.24: DLL scenario: US data rates

achievable with no and full crosstalk cancellation are plotted together with the target data rates in Fig. 7.24. More details on the DLL scenario can be found in Sec. 7.1.5.

7.2.3.3 Simulation Results

In this section, the simulation results are stated. Joint S-TLS+IWF is compared to S-JTLS without spectrum management and to IWF and S-JTLS when they are performed sequentially. The comparison criterion will be the percentage of users with achieved target data rates. It is given as the amount of users that have already achieved their desired data rates over the total number of users N in the binder in percent.

7.2.3.3.1 Required Run-Time Complexity

In Fig. 7.25 the run-time complexities which all three user groups require to fulfill their data rate requirements are compared. The sequential execution of IWF and S-JTLS shows suboptimal performance. 20% of the users are not able to reach their desired data rate even with full crosstalk cancellation. Iterative waterfilling optimizes the spectra for a full crosstalk situation and does not take into account, that interferences can be canceled on the receiver side. S-JTLS is able to fulfill all data rate targets at a run-time complexity of 25%. The best performance is achieved with Joint S-TLS+IWF as it only requires 20% run-time complexity to reach all data rate goals. It profits from the joint combination with IWF as it saves 5% of run-time complexity.

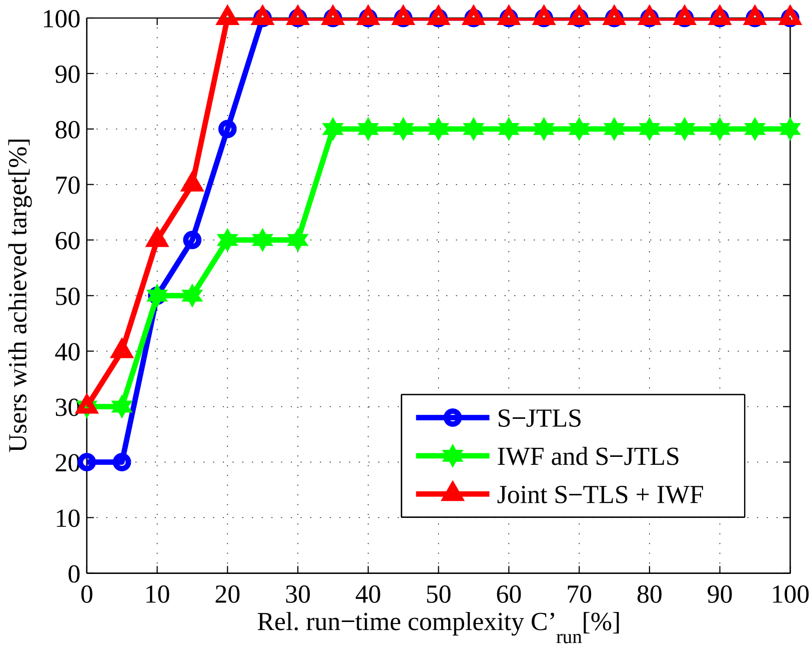


Figure 7.25: Comparison of needed complexity to fulfill the target data rates

Figure 7.26 shows the achievable data rates for $C'_{run} = 20\%$. Joint S-TLS+IWF achieves the data rate requirements for all lines. For line 8, 9 and 10 the data rate target is exceeded by around 5 Mbit/s. The same can be seen for line 2 and line 6. The rates of the remaining lines are close to the desired data rates. S-JTLS is not able to fulfill the data rate goals for lines 7 and 8. The sequential execution of IWF and S-JTLS does not allow users 1, 5, 6 and 7 to achieve their data rate targets.

During simulation the number of iterations of the outer loop of Joint S-TLS+IWF was counted. For the chosen scenario and the chosen target data rates the algorithms generally converges after a maximum of 4 iterations.

7.2.3.3.2 Run-Time Complexity Per Line for Target Data Rate Fulfillment

The performance of S-JTLS, Joint S-TLS+IWF and sequential execution of IWF and S-JTLS is analyzed for 25%, 20% and 30% and 100% of run-time complexity respectively in terms of achieved data rates and distribution of run-time complexity to the lines. For IWF and S-JTLS the performance for two complexities are analyzed as it never achieves all required rates.

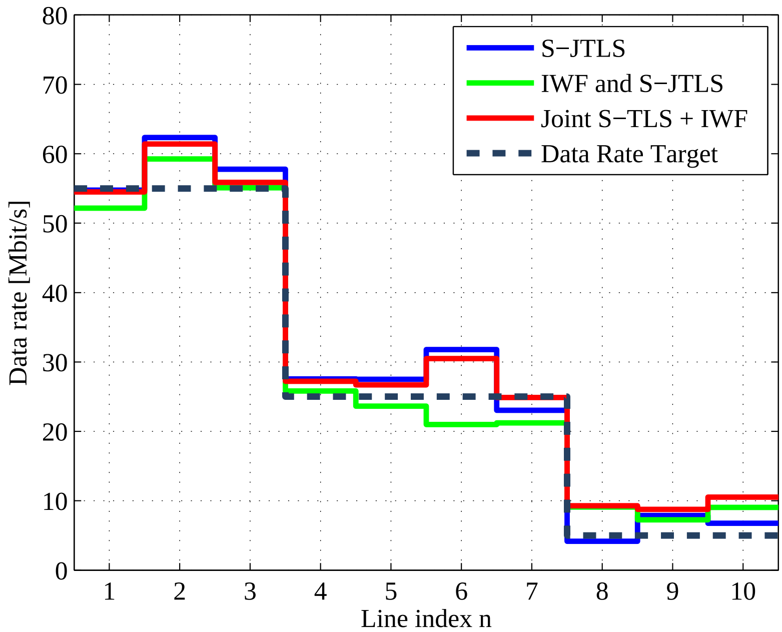
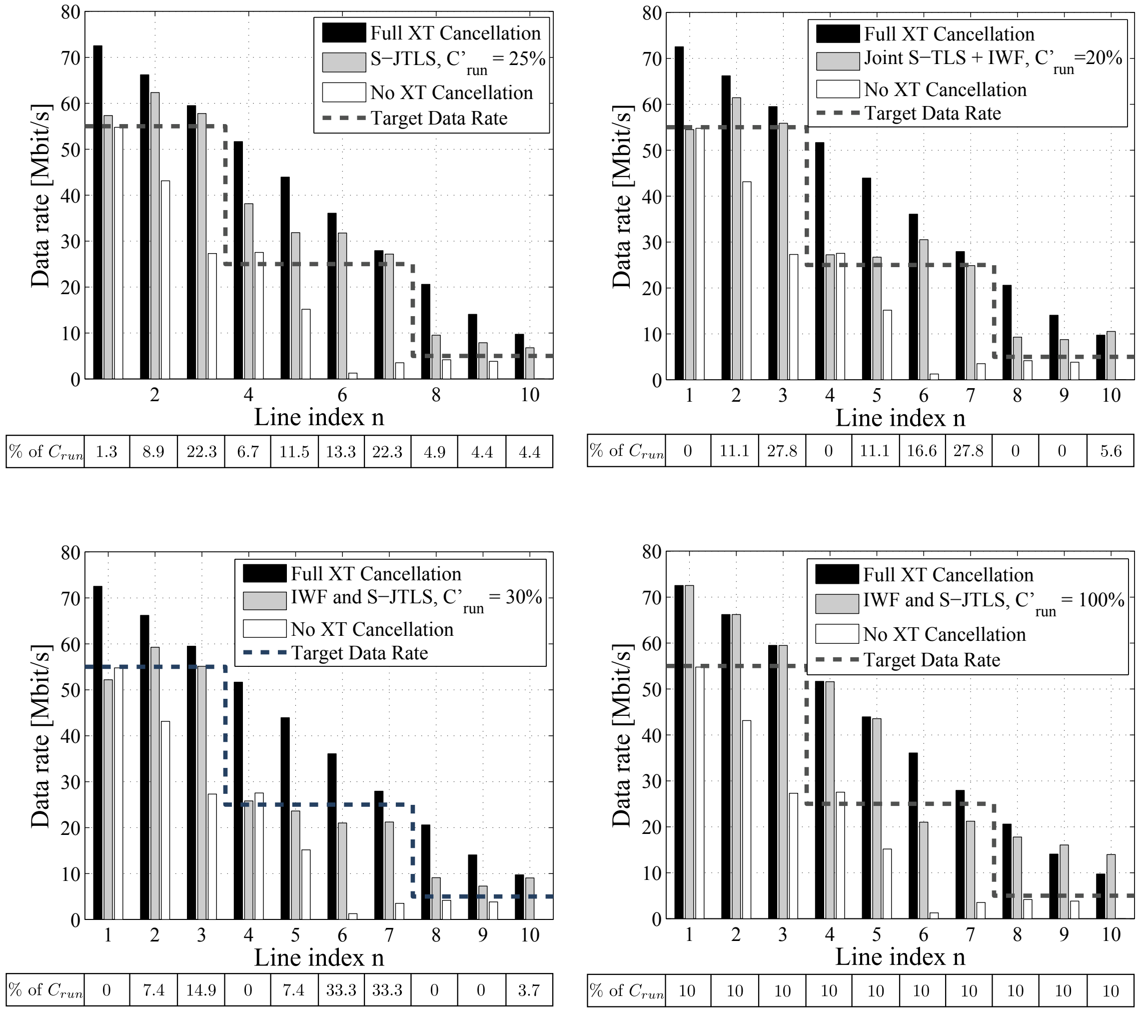


Figure 7.26: Comparison of achieved data rates for C'_{run} =20%

Figure 7.27 shows the achieved rates and the percentage of run-time complexity given to the lines in the binder when the data rate targets were first fulfilled. S-JTLS and Joint S-TLS+IWF both spend more than 20% of their resources to line 3 and 7. All long lines and line 1 require far less than 10% of the total available complexity. The target data rates are hit quite well for both algorithms.

Iterative waterfilling and S-JTLS sequentially performed distribute the run-time complexity in a comparable manner, but they are not able to fulfill the data rate demands at all. For an available complexity of 30%, line 1 and all long lines get no or only really low amounts of computational complexity. Line 6 and 7 get more than 30% of the computational resources, but their data rate requirements are not achieved yet. For $C'_{run} = 100\%$ the available run-time complexity is equally distributed to all lines. The long lines and line 4 and 5 achieve their maximum rate. But all other lines show suboptimal performance. The long lines are not able to achieve their highest data rates. The performance of line 6 and 7 did not change compared to an available run-time complexity of 30%.

Table 7.10 shows the absolute run-time complexity in terms of number of crosstalker-tone pairs selected for cancellation by the different selection algorithms to allow each user to fulfill his data rate requirements. When a line is fully canceled, the interferences of



% of C_{run}	1.3	8.9	22.3	6.7	11.5	13.3	22.3	4.9	4.4	4.4

% of C_{run}	0	11.1	27.8	0	11.1	16.6	27.8	0	0	5.6

% of C_{run}	0	7.4	14.9	0	7.4	33.3	33.3	0	0	3.7

% of C_{run}	10	10	10	10	10	10	10	10	10	10

Figure 7.27: Percentage of complexity distributed to the lines when all users achieved their data rate targets in upstream

10323 ($=$ 9 (crosstalkers) $\cdot$ 1147 (tones)) crosstalker-tone pairs are eliminated. Taking into account Table 7.10, the observed performance results can be explained. Joint S-TLS+IWF and S-JTLS obviously distribute the computational resources effectively amongst the lines in the binder as they require a comparably low amount of run-time complexity. Joint S-TLS+IWF is able to satisfy lines 1, 4, 8 and 9 without spending any computational resources on them. The complexity that Joint S-TLS+IWF gives to line 5 is slightly lower compared to what S-JTLS distributes to that line. All other lines are allocated more resources than by S-JTLS. The behavior of IWF performed sequentially with S-JTLS is especially interesting for the lines $6-10$. Line 6 and 7 already perform full crosstalk cancellation for

Table 7.10: Absolute run-time complexity per line to fulfill target data rates in US

line n	1	2	3	4	5	6	7	8	9	10
S-JTLS	340	2294	5735	1729	2966	3441	5735	1273	1147	1147
IWF and S-JTLS(30%)	0	2294	4588	0	2294	10323	10323	0	0	1147
IWF and S-JTLS(100%)	10323	10323	10323	10323	10323	10323	10323	10323	10323	10323
Joint S-TLS+IWF	0	2867	7168	0	2867	4301	7168	0	0	1433

an available complexity of 30%. These two lines as well as all long lines are not able to reach their data rate targets and their maximum achievable data rate even with full crosstalk cancellation.

7.2.3.3.3 Achievable Data Rates

In Fig. 7.28 the achievable data rates for relative computational complexities from 0 to 100% are compared for Joint S-TLS+IWF, S-JTLS and the sequential execution of IWF and S-JTLS. The left side shows the achievable data rates for lines 1 to 5. The right side displays them for lines 6 to 10.

S-JTLS and Joint S-TLS+IWF achieve all data rate targets for a run-time complexity of 25% and 20% respectively. It can be observed, that for both algorithms the resources are only allocated to the users with the highest gains after all lines achieved their data rate requirements. Comparing the two algorithms the biggest difference can be seen for lines 6 to 10. For those lines the desired target data rates can be achieved at a lower run-time complexity. They profit from the combination with IWF. For the long lines the performance is the same, except for line 1. For Joint S-TLS+IWF the achievable rate for line 1 decreases from 10% to 15% run-time complexity. This can be explained by the optimization the algorithm is doing. It decreases, but just gets closer to the target it has to achieve. When IWF and S-JTLS is performed sequentially, not all data rate requirements can be achieved due to the bad performance on lines 6 to 10. Line 6 and 7 are no longer able to achieve their target data rates. When the crosstalk on those lines is fully canceled, the remaining complexity is given to the lines with the highest gains.

7.2.3.3.4 Transmit Power Spectral Density

Fig. 7.29 illustrates the PSDs for all users. For JTLS the power allocation is not changed as no spectrum management is performed. That is why the PSD remains flat. For IWF and S-JTLS performed sequentially and Joint S-TLS+IWF, the resulting spectra are equal but the resulting data rates are not comparable. The figure shows that for line 1 to 5 the spectra change only slightly. The longer the line, the more change in power allocation. All tones are still used for transmission. For lines 6 to 10 the power is not allocated any more to the tones with indexes higher than 500. Tones with indexes higher than 500 are more influenced by noise and thus more vulnerable to crosstalk. The spectra of lines 6 to 10 move towards the lower frequencies. For Joint S-TLS+IWF the long lines can strongly benefit from IWF. The spectra of lines 1 to 5 nearly remain unchanged. These lines are not able to take big advantage of the joint procedure.

7.3 Summary

In this chapter, selection algorithms for partial crosstalk cancellation were presented. Their aim is to fulfill data rate requirements of all users in a binder and at the same time to minimize the needed run-time complexity by shifting the available complexity resources to where they are needed. The successive solutions S-LS, S-TS and S-JTLS were explained and analyzed. For further performance improvement IWF and S-JTLS was jointly combined in Joint S-TLS+IWF.

All proposed algorithms outperform the conventional solution JTLS in terms of needed run-time complexity. The computational resources are efficiently distributed. The joint combination with IWF is able to decrease the required run-time complexity.

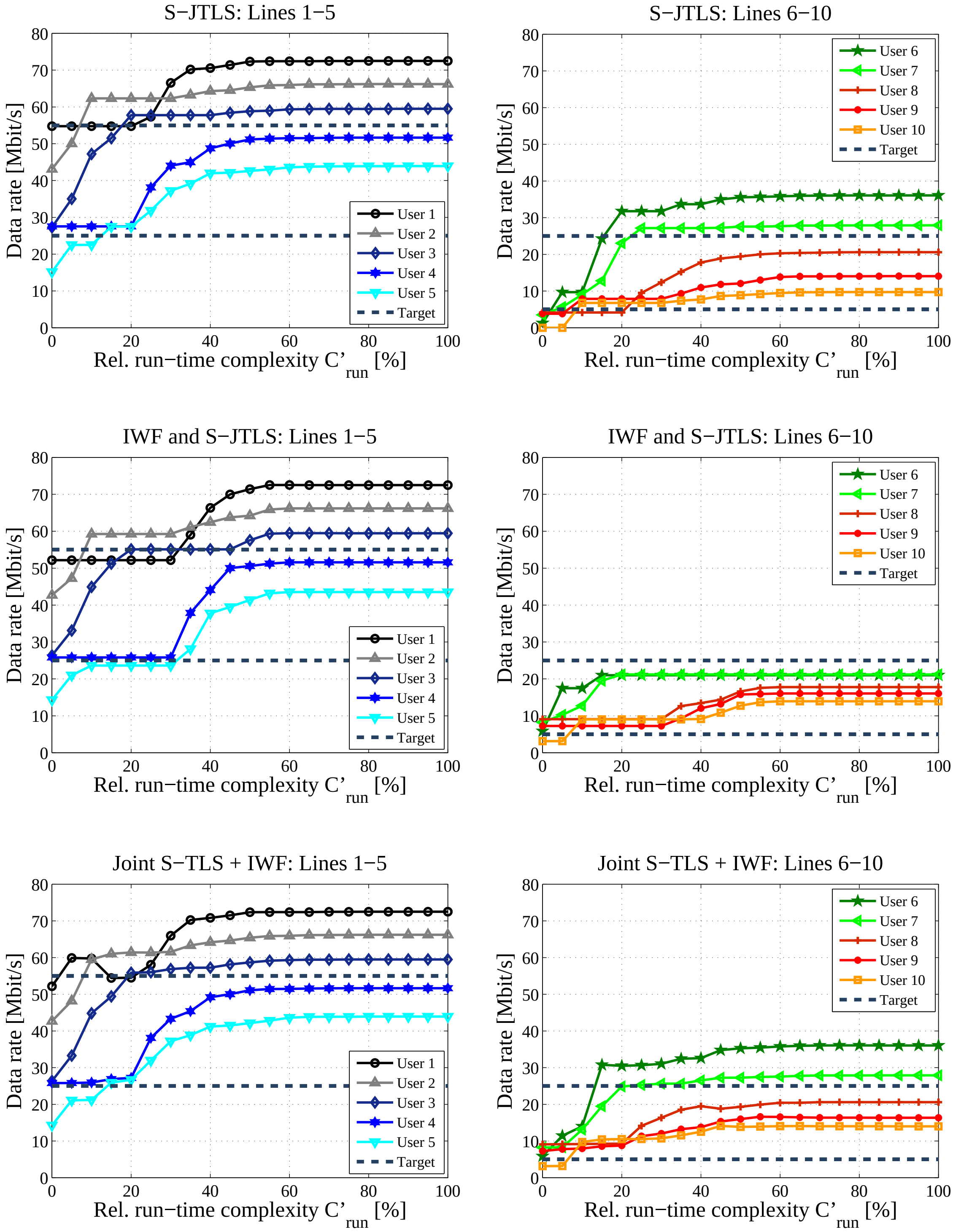


Figure 7.28: Achievable upstream data rates depending on available run-time complexity for S-JTLS, sequential execution of IWF and S-JTLS and Joint S-TLS+IWF

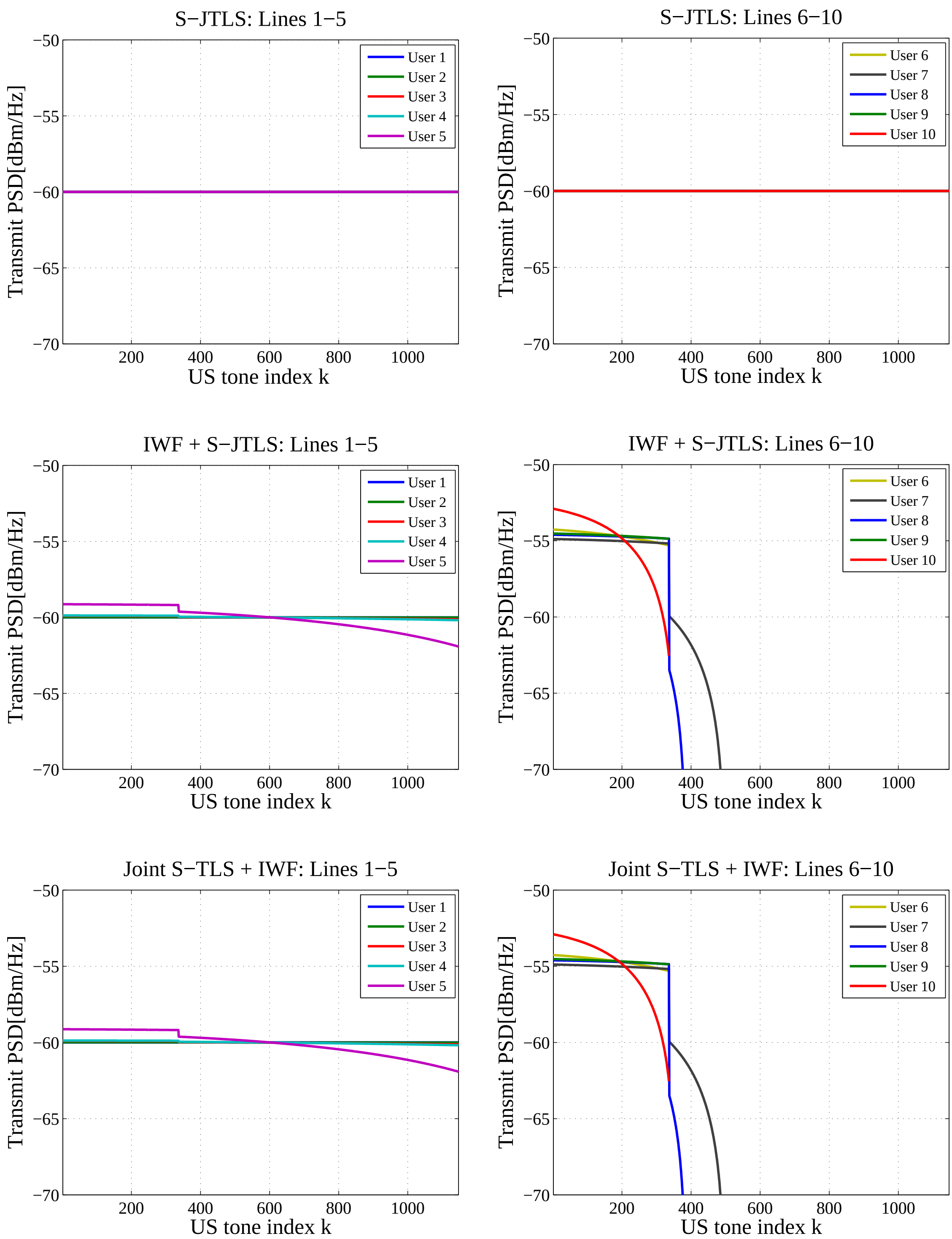


Figure 7.29: PSDs for S-JTLS, sequential execution of IWF and S-JTLS and Joint S-TLS+IWF

VIII

Conclusions

As a first contribution of this thesis, channel estimation and update procedures were proposed to measure the DSL channel with sufficient accuracy and to overcome data rate losses due to channel changes. The presented methods use orthogonal pilot sequences, which were transmitted in an initial channel estimation as well as continuously during the sync symbols. Due to the utilization of the sync symbol, any pilot overhead was avoided. For the channel adaptation in upstream, correlation was applied in the receiver. In downstream, feedback of the normalized error sample is required and therefore the channel coefficient update was calculated on the transmitter side using correlation. It was shown that both procedures achieve high data rate increases. In summary, the designed channel estimation and update procedures lead to a higher efficiency and increased performance figures in a VDSL2 system.

As a second and main contribution, partial crosstalk cancellation and precoding procedures were suggested, which allow all users in a DSL cable binder to achieve high data rate targets at a limited amount of available computational complexity. The users of a binder were split into three groups based on their distance to the CO. For each of the three groups, target data rates were defined.

First, three novel successive selection algorithms were presented. For high and low data rate demands of the different user groups the performance of the proposed algorithms S-LS, S-TS and S-JTLS was compared to the performance of conventional JTLS in upstream and downstream for a VDSL scenario with varying line lengths. In upstream, it has been shown that for high data rate targets S-JTLS and S-LS required 25% run-time complexity, S-TS required 40% and JTLS showed worst performance with needing 50% run-time complexity. Also for low data rate targets, all successive algorithms outperformed JTLS, which needed 40% run-time complexity. S-JTLS used 15%, where S-LS and S-TS required 20% and 30% run-time complexity respectively. It was shown for upstream transmission that the successive selection algorithms distributed the available complexity resources to the lines which required them most. Only when all users achieved their data rate goals and there was still computational complexity available, they spent it on the lines where most rate was gained.

Downstream performance results also showed, that all successive algorithms outperform JTLS. S-JTLS required 65% and 25% run-time complexity for high and low data rate target. S-LS needed 65% and 50% run-time complexity and S-TS used 75% and 50% computational resources when all data rate requirements were achieved. JTLS needed 85% and 70% run-time complexity for high and low data rate targets. It has been observed that also in downstream, computational resources were distributed effectively, though the algorithms showed suboptimal behavior due to inaccuracies of the data rate estimation within the procedure.

In summary, it can be stated that all lines in a binder were able to achieve their data rate goals at a limited computational complexity with the successive selection algorithms. The best performance was shown by S-JTLS. Though not having been fully accurate in some transmission scenarios all successive algorithms outperformed conventional JTLS in terms of needed complexity to achieve the desired data rates.

Finally, a procedure was presented which combines partial crosstalk cancellation and spectrum management in upstream to fulfill data rate requirements of all users in a binder. Successive joint tone-line selection was used, which focuses on data rate fulfillment instead of capacity maximization. Joint successive tone-line selection and iterative waterfilling was explained, which simultaneously selects the transmit spectra of the users and the canceled crosstalkers for each line. An upstream VDSL scenario with different line lengths was considered. It was shown that with Joint S-TLS+IWF, data rate requirements of all users in a binder are supported. Additionally, compared to S-JTLS and the sequential execution of IWF and S-JTLS savings in needed computational complexity to reach all data rate requirements were achieved. The joint combination of S-JTLS and IWF required 20% run-time complexity to fulfill all data rate requirements of all users in the binder with high data rate demands. S-JTLS required 5% more computational resources compared to Joint S-TLS+IWF. For the chosen transmission scenario the sequential execution of IWF and S-JTLS without any joint processing did not perform well even for full crosstalk cancellation. The target data rates could not be achieved. It was shown that Joint S-TLS+IWF shapes the transmit spectra of the long lines to allow them to achieve better performance. The combination with partial cancellation leads to a better distribution of available computational complexity.

Appendix A

A.1 Derivations of Waterfilling Solution

A.1.1 Rate Maximization

Optimization problem:

$$\max b(P_k, b_k) = \max \left\{ \sum_{k=1}^{K} b_k \right\} \tag{A-1}$$

$$= \max \left\{ \sum_{k=1}^{K} \log_2 \left(1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \sigma^2} \right) \right\} \tag{A-2}$$

$$\text{subject to: } \sum_{k=1}^{K} P_k = P_{max} \tag{A-3}$$

Maximize the number of loaded bits b for a fixed given power P_{max} by adapting the power allocation and the bit allocation on the tones. P_k is the power on tone k. b_k is the number of bits loaded on tone k. K is the total number of tones.
Build cost function with Lagrange multiplier:

$$L(\lambda, P_1, \ldots, P_k) = \sum_{k=1}^{K} b_k - \lambda \sum_{k=1}^{K} P_k \tag{A-4}$$

$$= \sum_{k=1}^{K} \log_2 \left(1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \sigma^2} \right) - \lambda \sum_{k=1}^{K} P_k. \tag{A-5}$$

Differentiate L with respect to P_k to find the maximum:

$$\frac{\partial L}{\partial P_k} = \frac{1}{\ln 2} \cdot \frac{1}{1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \cdot \sigma^2}} \cdot \frac{|H_k|^2}{\Gamma \cdot \sigma^2} - \lambda \tag{A-6}$$

$$= \frac{1}{\ln 2} \cdot \frac{|H_k|^2}{\Gamma \cdot \sigma^2 + |H_k|^2 \cdot P_k} - \lambda \tag{A-7}$$

$$= 0 \tag{A-8}$$

$$\frac{1}{\ln 2}\cdot\frac{1}{\frac{\Gamma\cdot\sigma^2}{|H_k|^2}+P_k}-\lambda=0 \tag{A-9}$$

$$\Leftrightarrow P_k=\frac{1}{\lambda\cdot\ln 2}-\frac{\Gamma\cdot\sigma^2}{|H_k|^2} \tag{A-10}$$

Use the constraint to replace the Lagrange multiplier:

$$\sum_{k=1}^{K}P_k=P_{max} \tag{A-11}$$

$$\Leftrightarrow\sum_{k=1}^{K}\left(\frac{1}{\lambda\cdot\ln\ 2}-\frac{\Gamma\cdot\sigma^2}{|H_k|^2}\right)=P_{max} \tag{A-12}$$

$$\Leftrightarrow\sum_{k=1}^{K}\frac{1}{\lambda\cdot\ln 2}-\sum_{k=1}^{K}\frac{\Gamma\cdot\sigma^2}{|H_k|^2}=P_{max} \tag{A-13}$$

$$\Leftrightarrow\frac{K}{\lambda\cdot\ln 2}=P_{max}+\sum_{k=1}^{K}\frac{\Gamma\cdot\sigma^2}{|H_k|^2} \tag{A-14}$$

$$\Leftrightarrow\lambda=\frac{K}{\ln 2\left(P_{max}+\sum_{k=1}^{K}\frac{\Gamma\cdot\sigma^2}{|H_k|^2}\right)} \tag{A-15}$$

Replace the Lagrange multiplier Eq. A-10:

$$P_k=\frac{1}{\frac{K}{\ln 2\left(P_{max}+\sum_{k=1}^{K}\frac{\Gamma\cdot\sigma^2}{|H_k|^2}\right)}\cdot\ln 2}-\frac{\Gamma\cdot\sigma^2}{|H_k|^2} \tag{A-16}$$

$$=\frac{1}{K}\left(P_{max}+\sum_{k=1}^{K}\frac{\Gamma\cdot\sigma^2}{|H_k|^2}\right)-\frac{\Gamma\cdot\sigma^2}{|H_k|^2} \tag{A-17}$$

$$=\mu_{ra}-\frac{\Gamma\cdot\sigma^2}{|H_k|^2}. \tag{A-18}$$

A.1.2 Power Minimization

Optimization problem:

$$\min P(b_k,P_k)=\min\left\{\sum_{k=1}^{K}P_k\right\} \tag{A-19}$$

$$\text{subject to: }\sum_{k=1}^{K}\log_2\left(1+\frac{|H_k|^2\cdot P_k}{\Gamma\cdot\sigma^2}\right)=b^{(T)}. \tag{A-20}$$

Minimize the power P for a fixed given target number of loaded bits $b^{(T)}$ by adapting the power allocation and the bit allocation on the tones. P_k is the power on tone k. b_k is the number of bits loaded on tone k. There exist K tones in total.

Build cost function with Lagrange multiplier:

$$L(\lambda, P_1, \ldots, P_k) = \sum_{k=1}^{K} P_k - \lambda \sum_{k=1}^{K} \log_2 \left(1 + \frac{|H_k|^2 \cdot P_k}{\Gamma \cdot \sigma^2}\right) \tag{A-21}$$

Differentiate with respect to P_k to find the maximum:

$$\frac{\partial L}{\partial P_k} = 1 - \lambda \cdot \frac{1}{\left(\frac{\Gamma \cdot \sigma^2}{|H_k|^2} + P_k\right) \ln 2} = 0 \tag{A-22}$$

$$\Leftrightarrow \lambda \cdot \frac{1}{\left(\frac{\Gamma \cdot \sigma^2}{|H_k|^2} + P_k\right) \ln 2} = 1 \tag{A-23}$$

$$\Leftrightarrow P_k = \frac{\lambda}{\ln 2} - \frac{\Gamma \sigma^2}{|H_k|^2} \tag{A-24}$$

Use the constraint to replace the Lagrange multiplier:

$$b^{(T)} = \sum_{k=1}^{K} \log_2 \left(1 + \frac{\left(\frac{\lambda}{\ln 2} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2}\right) \cdot |H_k|^2}{\Gamma \cdot \sigma^2}\right) \tag{A-25}$$

$$= \log_2 \prod_{k=1}^{K} \left(1 + \frac{\frac{\lambda |H_k|^2}{\ln 2} - \Gamma \sigma^2}{\Gamma \cdot \sigma^2}\right) \tag{A-26}$$

$$= \log_2 \prod_{k=1}^{K} \left(1 + \frac{\lambda |H_k|^2}{\ln 2 \cdot \Gamma \cdot \sigma^2} - 1\right) \tag{A-27}$$

$$2^{b^{(T)}} = \prod_{k-1}^{K} \left(\frac{\lambda |H_k|^2}{\ln 2 \cdot \Gamma \cdot \sigma^2}\right) \tag{A-28}$$

$$= \lambda^K \prod_{k-1}^{K} \left(\frac{|H_k|^2}{\ln 2 \cdot \Gamma \cdot \sigma^2}\right) \tag{A-29}$$

$$=> \lambda = \left(\frac{2^{b^{(T)}}}{\prod_{k=1}^{K} \left(\frac{|H_k|^2}{\ln 2 \cdot \Gamma \cdot \sigma^2}\right)}\right)^{1/K} \tag{A-30}$$

Replace the Lagrange multiplier Eq. A-24:

$$P_k = \Gamma \cdot \left(\frac{2^{b^{(T)}}}{\prod_{k=1}^{K} \left(\frac{|H_k|^2}{\sigma^2} \right)} \right)^{1/K} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \quad \text{(A-31)}$$

$$= 2^{\frac{1}{K}\left[b^{(T)} - \sum_{k=1}^{K} \log_2 \left(\frac{|H_k|^2}{\Gamma \cdot \sigma^2} \right) \right]} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \quad \text{(A-32)}$$

$$= \mu_{pa} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2} \quad \text{(A-33)}$$

A.2 Implementation of Waterfilling Algorithms

A.2.1 Rate-Adaptive Waterfilling Algorithm

Sequence of the algorithm:

1. Start with a flat power allocation.
2. Calculate the channel gains $g_k = |H_k|^2/\sigma^2$ and sort them from largest to smallest. Initialize $k = 1, \dots, K$ and $K^* = K$.
3. Calculate μ_{ra} and P_{K^*}.
4. If $P_{K^*} \leq 0 :\ K^* \to K^* - 1$. Go to step 3.
   Else: go to the next step.
5. Set $P_k = \mu_{ra} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2}$ and $b_k = \log_2 \left(1 + \frac{P_k \cdot |H_k|^2}{\Gamma \sigma^2} \right)$
   for $k = 1 \dots K^*$.

A.2.2 Power-Adaptive Waterfilling Algorithm

Sequence of the algorithm:

1. Start with a flat power allocation.
2. Calculate the channel gains $g_k = |H_k|^2/\sigma^2$ and sort them from largest to smallest. Initialize $k = 1, \dots, K$ and $K^* = K$.

3. Calculate μ_{pa} and P_{K^*}.

4. If $P_{K^*} \leq 0 : K^* \rightarrow K^* - 1$. Go to step 3.
   Else: go to the next step.

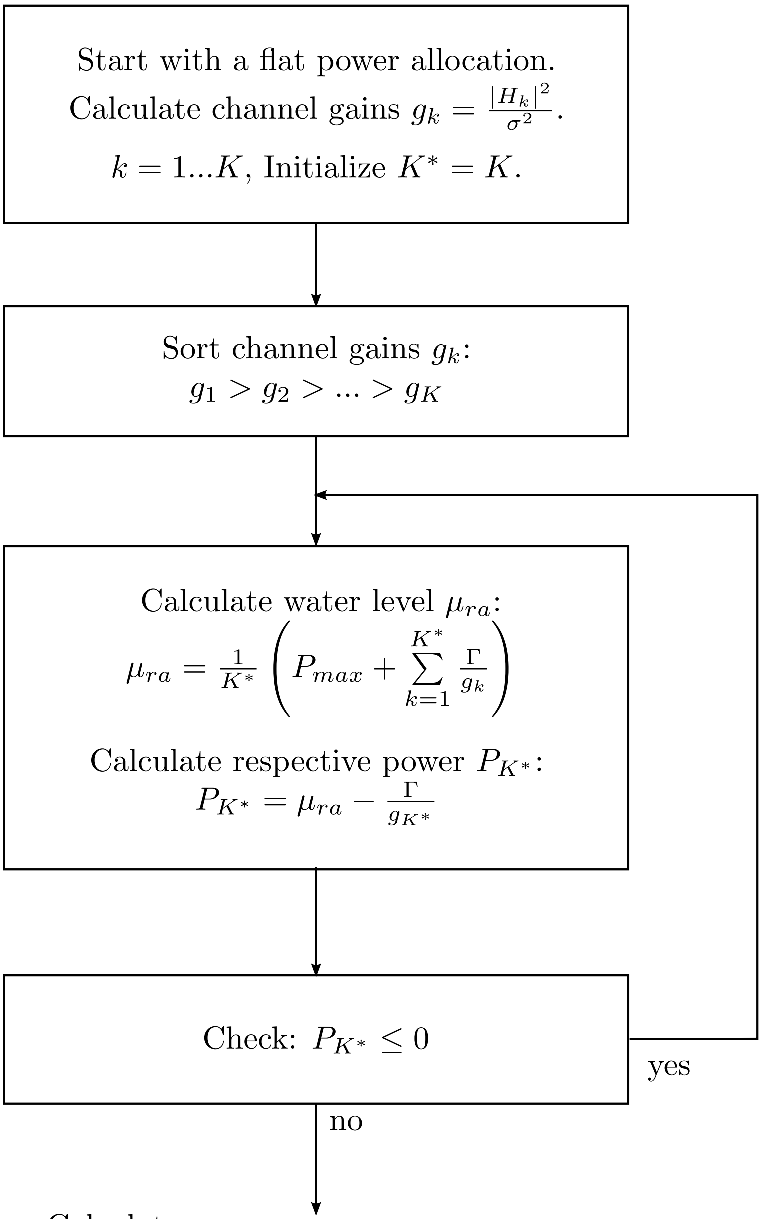
5. Set $P_k = \mu_{pa} - \frac{\Gamma \cdot \sigma^2}{|H_k|^2}$ and $b_k = \log_2 \left(1 + \frac{P_k \cdot |H_k|^2}{\Gamma \sigma^2}\right)$
   for $k = 1 \dots K^*$.

There exist two implementations for waterfilling algorithms which minimize the power. Power-adaptive waterfilling finds the minimum power $P(b_k, P_k)$ which is needed to achieve a given number of loaded bits and maximizes the margin at the same time. To do margin maximization, the calculation of the margin

$$\gamma_{max} = \frac{P}{\sum_{k=1}^{K^*} P_k} \tag{A-34}$$

has to be added to step 5.

For fixed-margin waterfilling, the power is also minimized but the margin is set to a fixed given value.



Calculate:

$$P_k = \begin{cases} \mu_{ra} - \frac{\Gamma}{g_k}, & k = 1...K^* \\ 0, & \text{else} \end{cases}$$

$$b_k = \begin{cases} \log_2\left(1 + \frac{P_k \cdot g_k}{\Gamma}\right), & k = 1...K^* \\ 0, & \text{else} \end{cases}$$

Unsort subchannels.

Figure A.1: Flow chart for rate-adaptive waterfilling

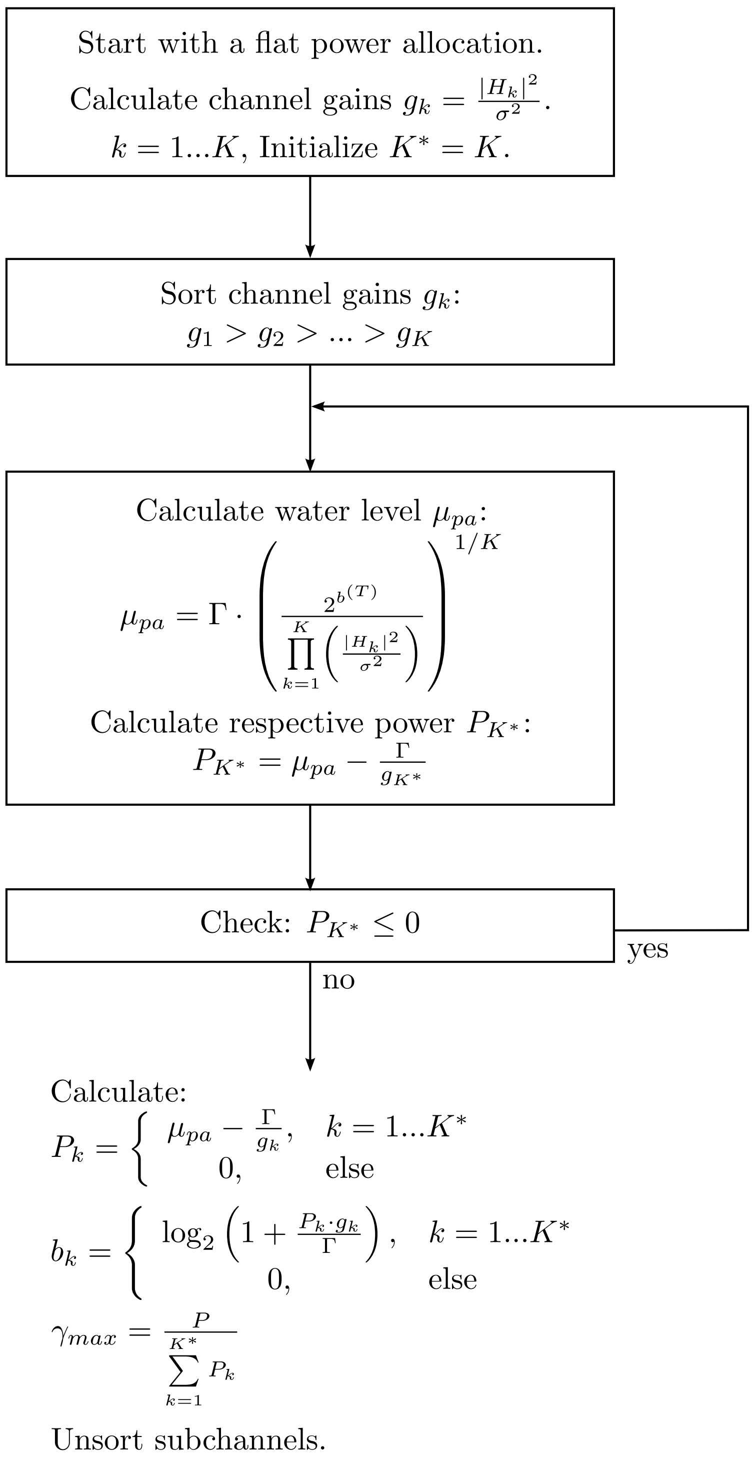


Figure A.2: Flow chart for power-adaptive waterfilling

Appendix B

The performance analysis of the algorithms presented in Chapter 7 is based on a DLL transmission scenario. The scenario is shortly described in Sec. 7.1.5. As additional information the channel and crosstalk transfer functions, the PSD and the Signal-to-Interference Ratio (SIR) of all 10 lines in the considered binder are presented here.

Figures B.2 to B.5 show the channel and crosstalk transfer coefficients on all used tones on the left side and the receive PSDs on the right side for all users in the binder. In the calculation of the receive PSDs, a transmission power of -60 dBm/Hz was assumed. Figure B.1 additionally shows the resulting SIR experienced by all lines in the binder. In all plots five frequency bands can be observed. Starting at the left, the first, third and fifth band are used for downstream transmission. Band 2 and 4 are available for upstream.

Figure B.1 shows that in upstream transmission strong near-far effects are present in the binder as upstream crosstalk strength strongly varies from line to line. User 10, who has the largest distance to the CO, is heavily affected by interferences whereas line 1, which is closest to the CO, has good channel properties. Generally it can be observed in upstream that the overall crosstalk influence on a line increases with line length. In the considered scenario line 6 is the only exception as its SIR is also quite low. In downstream, the crosstalk influences on the lines are more homogeneous. The difference in SIR is around 15 dB from the best to the worst line. User 8 has the best channel. User 1 has worst transmission properties.

The channel and crosstalk transfer functions and the receive PSDs show the strength of the individual crosstalkers on the different lines. It can be observed that in upstream there are lines with one or two dominant crosstalkers (e.g. line 6, line 10) and lines with several crosstalkers of comparable strength (e.g. line 4, line 1). In downstream crosstalk strength is again more homogeneous, but there are also lines with some dominant crosstalkers (e.g. line 1) and lines with several crosstalkers of comparable strength (e.g. line 5, line 10).

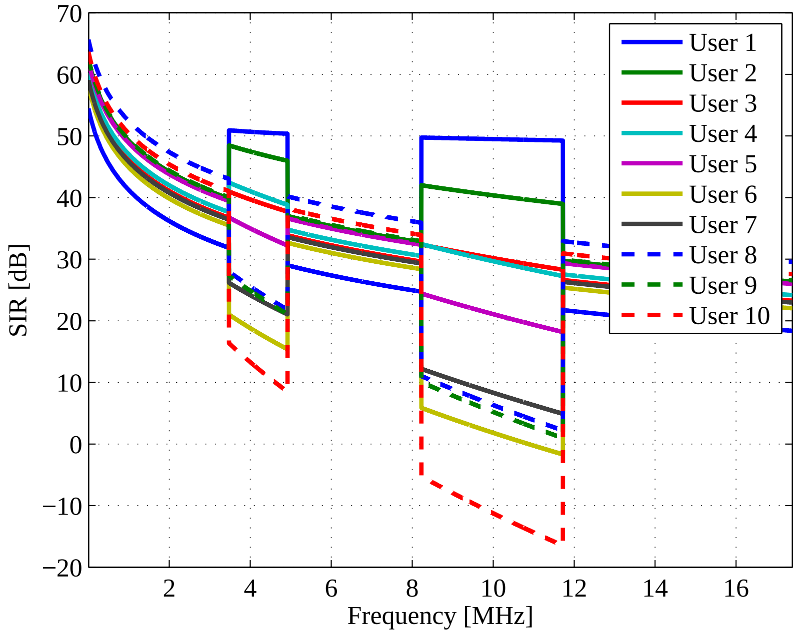


Figure B.1: SIRs for all lines in the binder

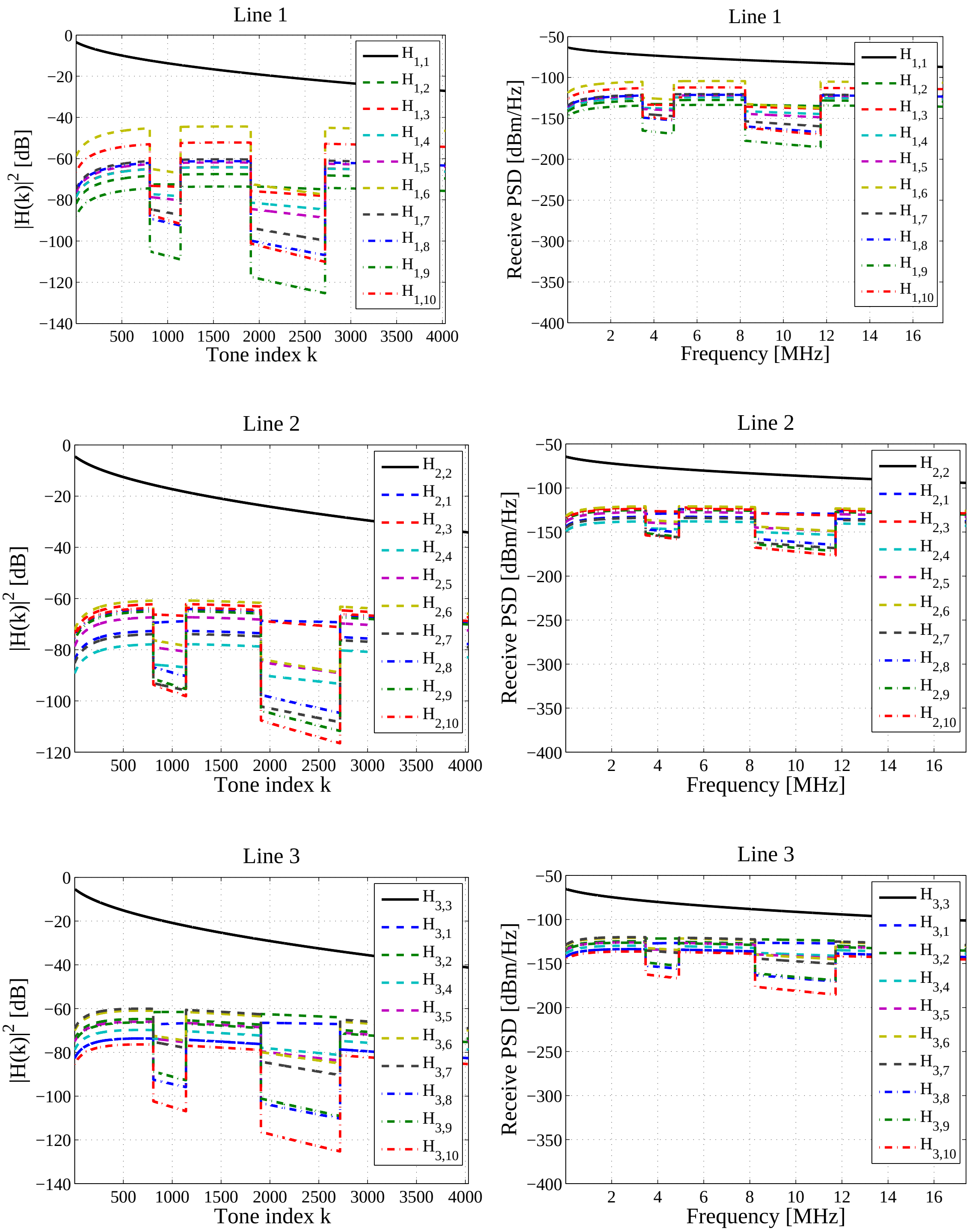


Figure B.2: Channel and crosstalk transfer functions (left) and PSDs (right) for line 1 to 3

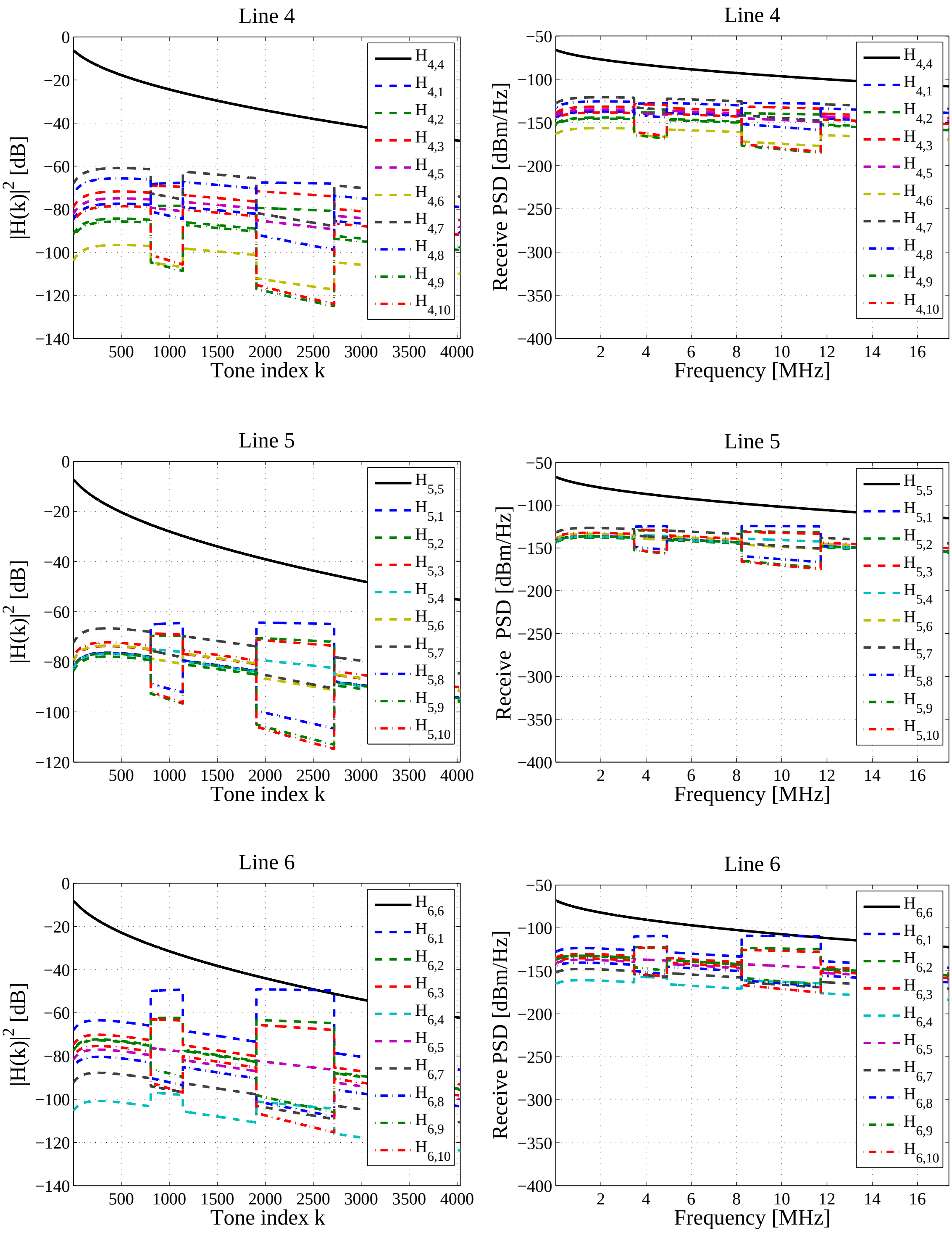


Figure B.3: Channel and crosstalk transfer functions (left) and PSDs (right) for line 4 to 6

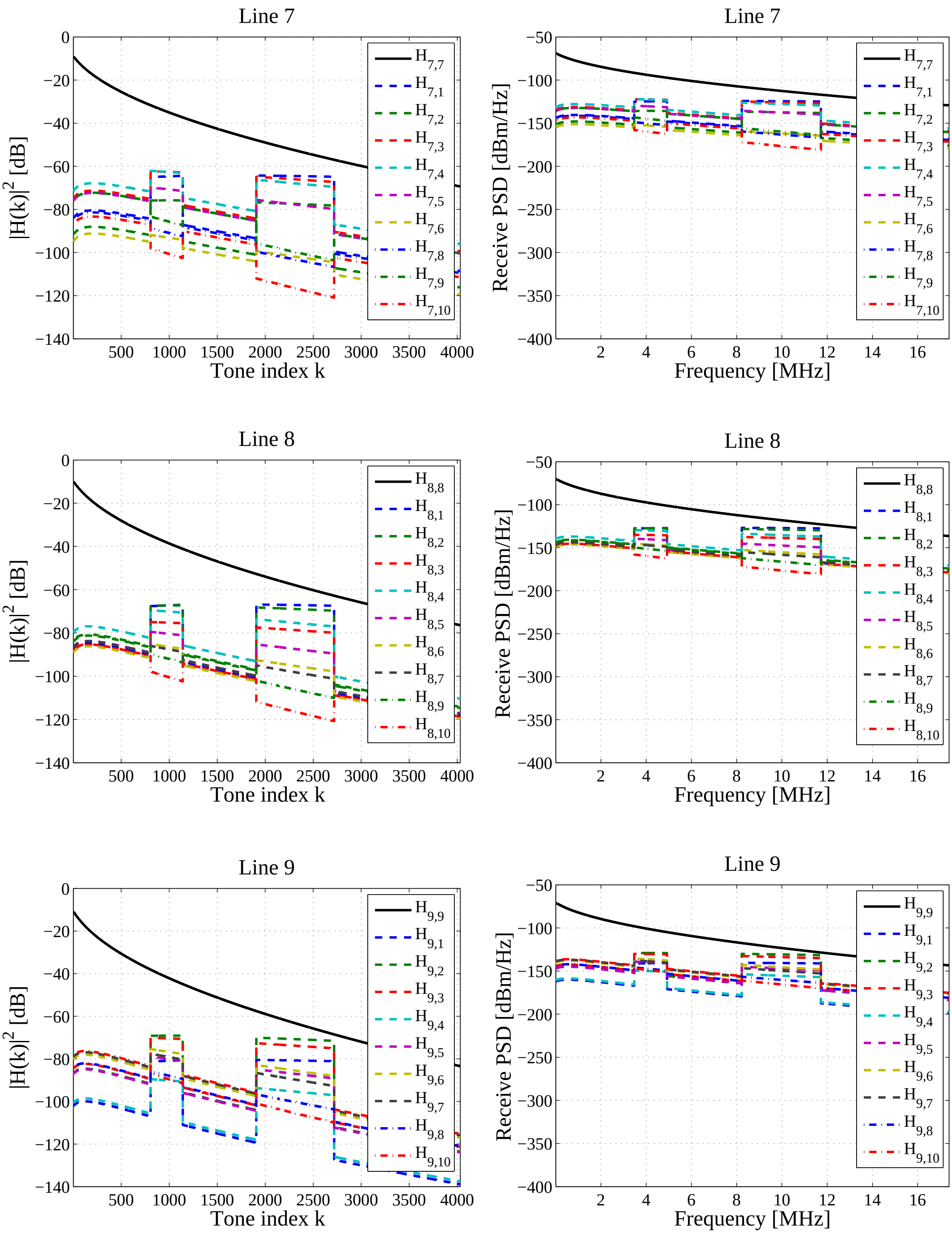


Figure B.4: Channel and crosstalk transfer functions (left) and PSDs (right) for line 7 to 9

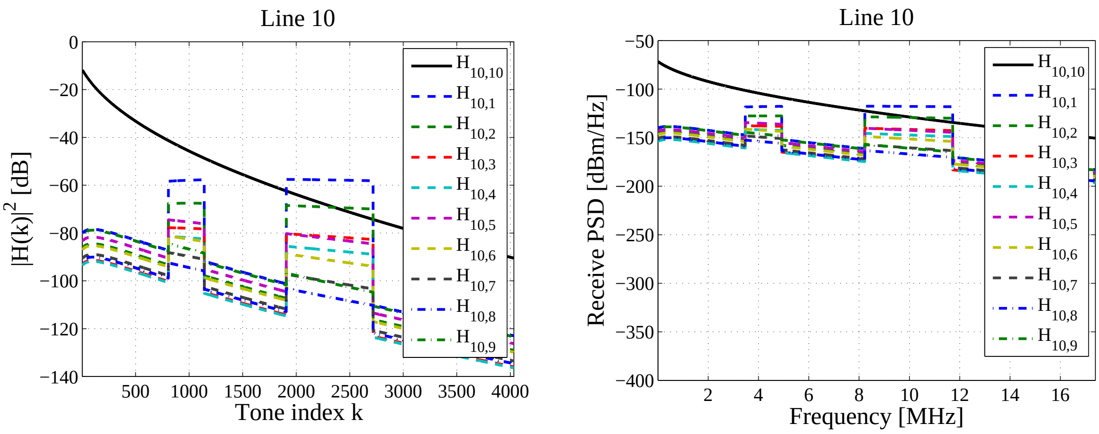


Figure B.5: Channel and crosstalk transfer functions (left) and PSDs (right) for line 10

C

Abbreviations and Symbols

C.1 Abbreviations

ADSL	Asymmetric Digital Subscriber Line
ASB	Autonomous Spectrum Balancing
AWGN	Additive White Gaussian Noise
CO	Central Office
CPE	Customer Premises Equipment
CSI	Channel State Information
CWDD	Column-Wise Diagonal Dominant
DAB	Digital Audio Broadcasting
DFC	Decision Feedback Canceler
DFE	Decision Feedback Equalizer
DFT	Discrete Fourier Transform
DLL	Distributed Line Length
DMT	Discrete Multi-Tone Transmission
DS	Downstream
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DSM	Dynamic Spectrum Management

DVB	Digital Video Broadcasting
ECH	Echo Cancellation Hybrid
FDD	Frequency Division Duplexing
FEXT	Far-End Crosstalk
FFT	Fast Fourier Transform
HDSL	High Data Rate Digital Subscriber Line
HDTV	High Definition Television
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
ISB	Iterative Spectrum Balancing
ISDN	Integrated Services Digital Network
ISI	Intersymbol Interferences
IWF	Iterative Waterfilling
Joint S-TLS+IWF	Joint Successive Tone-Line Selection and Iterative Waterfilling
JTLS	Joint Tone-Line Selection
LMS	Least Mean Squares
LTE	Long Term Evolution
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Square Error
NEXT	Near-End Crosstalk
OFDM	Orthogonal Frequency Division Multiplexing
ONU	Optical Network Unit

OSB	Optimum Spectrum Balancing
PA	Power-Adaptive
PBO	Power Back-Off
POTS	Plain Old Telephone Service
PSD	Power Spectral Density
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
RA	Rate-Adaptive
RFI	Radio Frequency Ingress
RT	Remote Terminal
RWDD	Row-Wise Diagonal Dominant
Rx	Receiver
SDSL	Symmetric Digital Subscriber Line
SIC	Successive Interference Cancellation
S-JTLS	Successive Joint Tone-Line Selection
S-LS	Successive Line Selection
SINR	Signal-to-Interference and Noise Ratio
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
SSM	Static Spectrum Management
S-TS	Successive Tone Selection
TDD	Time Division Duplexing

THP	Tomlinson-Harashima Precoder
Tx	Transmitter
UPBO	Upstream Power Back-Off
US	Upstream
VDSL	Very High Speed Digital Subscriber Line
VDSL2	Very High Speed Digital Subscriber Line 2
XT	Crosstalk
ZF	Zero-Forcing
FEQ	Frequency Domain Equalizer

C.2 Symbols

$\mathbf{A}$	ABCD matrix of a DSL loop
b	Number of loaded bits per symbol
$b^{(T)}$	Constraint on number of loaded bits
b_k	Number of bits loaded on tone k
b_k^+	Number of bits loaded on tone k for positive-only power loading
B	System bandwidth
c_{max}	Maximum number of crosstalkers per tone that can be canceled in the entire binder with S-LS
c_n	Number of crosstalkers that can currently be canceled for user n in S-LS
C_{init}	Initialization complexity
C_k	Capacity on tone k

C_{run}	Run-time complexity available for crosstalk reduction for the entire binder
C'_{run}	Relative run-time complexity available for crosstalk reduction in the entire binder
d	Line length
$d_{\text{coupling}}^{(n,m)}$	Coupling length between line n and line m
$d_n(i)$	Index of user n's crosstalker-tone pair with ith largest gain
$e_k^{(n)}(t)$	Normalized complex error for user n on tone k at time t
$\mathbf{E}_k(t)$	Error matrix on tone k at time instant t
f	Frequency
f_S	DMT symbol rate
g_k	Channel gain on tone k
$g_{n,k}^{TS}$	Bit gain of user n on tone k for S-TS
$g_n^{JTLS}(m,k)$	Capacity gain of crosstalker-tone pair (m,k) for user n on tone k for S-JTLS
gn_k^n	Channel gain factor of user n on tone k in Joint S-TLS+IWF
$H(f,d)$	Channel transfer function at frequency f and line length d
$H^{(n,m)}(f,d)$	Channel transfer function from line m to line n at frequency f and line length d
$H_{99}^{(n,m)}(f,d)$	99% worst-case channel transfer function from line m to line n at frequency f and line length d
H_k	Direct channel transfer coefficient on the kth tone
$H_k^{(n,m)}$	Channel transfer coefficient from line m to line n on tone k
$\hat{H}_k$	Estimated direct channel transfer coefficient on the kth tone

$\hat{H}_k^{(n,m)}$	Estimated channel transfer coefficient from line m to line n on tone k
$\mathbf{H}_k$	Channel transfer matrix on tone k
$\mathbf{H}_{k,diag}$	Diagonal matrix containing the direct channel transfer coefficients on tone k
$\mathbf{H}_{k,norm}$	Normalized crosstalk channel transfer matrix on tone k
$\hat{\mathbf{H}}_k$	Estimated channel transfer matrix on tone k
$\hat{\mathbf{H}}_{k,diag}$	Diagonal matrix containing the estimated direct channel transfer coefficients on tone k
$\hat{\mathbf{H}}_{k,norm}$	Estimated normalized crosstalk channel transfer matrix on tone k
$\bar{\mathbf{H}}_k^n$	Partial channel matrix used to obtain partial canceler coefficients for user n on tone k
$\bar{\mathbf{H}}_k^{pre,m}$	Partial channel matrix used to obtain partial precoder coefficients for user m on tone k
$\mathbf{I}_x$	Identity matrix of size $x \times x$
k	Tone index
$k_n(i)$	Index of user n's tone with ith largest bit gain for S-TS
K	Number of usable DMT tones
$\bar{K}$	Number of DMT tones
K^*	Number of tones which are loaded in the waterfilling procedure
K_{DS}	Number of usable DMT tones in downstream
K_u	Number of used tones (K_{US} in upstream, K_{DS} in downstream)
K_{US}	Number of usable DMT tones in upstream
l	Pilot symbol index
L	Length of orthogonal pilot sequence

m	Crosstalker index
$m_{n,k}(i)$	Index of user n's ith largest crosstalker on tone k
$\mathbb{M}_k^n$	Set of interferers canceled for line n on tone k
$\mathbb{M}_{QoS,k}^n$	Set of interferers canceled for line n on tone k to fulfill data rate constraints
$\mathbb{M}_{QoS,k}^{n,LS}$	Set of interferers canceled for line n on tone k to fulfill data rate constraints with S-LS
$\mathbb{M}_{QoS,k}^{n,TS}$	Set of interferers canceled for line n on tone k to fulfill data rate constraints with S-TS
$\mathbb{M}_{QoS,k}^{n,JTLS}$	Set of interferers canceled for line n on tone k to fulfill data rate constraints with S-JTLS
n	Line index
$n_k^{(n)}$	Additive noise contribution on line n on tone k
$n_{m,k}(i)$	Index of ith hardest sufferer from user m's interferences on tone k
N	Number of lines in the cable binder
N_k	AWGN contribution on the kth tone
N_{sf}	Number of DMT symbols per superframe
$\mathbf{n}_k$	Noise signal vector on tone k
$\tilde{\mathbf{n}}_k$	Scaled noise signal vector on tone k
$\mathbf{N}_k$	Noise matrix on tone k
$\tilde{\mathbf{N}}_k$	Scaled noise matrix on tone k
$\mathbb{N}_k^m$	Set of users who want to be protected against interference from transmitter m on tone k
$\mathbb{N}_{QoS,k}^m$	Set of users who want to be protected against interference from transmitter m on tone k when data rate constraints are considered

p_{max}	Maximum number of crosstalker-tone pairs that can be canceled for an entire binder with S-JTLS
p_n	Number of currently canceled crosstalker-tone pairs in S-JTLS for user n
$p_{n,k}$	Number of extra observation lines for line n on tone k
P	Transmit power
P_e	Bit error rate
P_k	Transmit power on tone k
P_k^{mask}	Spectral mask on tone k
P_k^n	Transmit power of line n on tone k
$P_k^{n,mask}$	Spectral mask for user n on tone k
P_k^+	Positive-only power loading per tone
$P_k^{+,mask}$	Positive-only power loading per tone with spectral mask limitations
P_{max}	Maximum available transmit power
P_{max}^n	Maximum available transmit power for line n
$\mathbf{P}_n$	PSD of user n
$r^{(n,l)}$	lth transmitted pilot symbol of user n
$r_k^{(n)}(t)$	Transmitted pilot symbol of user n on tone k at time instant t
$r_k^{(n,l)}$	lth transmitted pilot symbol of user n on tone k
$\hat{r}_k^{(n)}(t)$	lth normalized received pilot symbol of user n on tone k
$\tilde{r}_k^{(n)}(t)$	Received pilot symbol of user n on tone k at time instant t
$\tilde{r}_k^{(n,l)}$	lth received pilot symbol of user n on tone k
R	Data rate

R_n	Data rate on line n
$R_n^{(I)}$	Initial data rate on line n
$R_n^{(T)}$	Data rate target on line n
$R_{n,LS}^{(C)}(c_n)$	Current data rate for user n in S-LS for c_n canceled crosstalkers per tone
$R_{n,TS}^{(C)}(t_n)$	Current data rate for user n in S-TS for t_n fully canceled tones
$R_{n,JTLS}^{(C)}(p_n)$	Current data rate for user n in S-JTLS for p_n canceled crosstalker-tone pairs
$\mathbf{R}$	Matrix of pilot sequences
$\mathbf{R}_k(t)$	Matrix of transmitted pilot sequences on tone k at time instant t
$\hat{\mathbf{R}}_k(t)$	Matrix of normalized received pilot sequences on tone k at time instant t
SNR_k	Signal-to-noise ratio on tone k
$SINR_k^n$	Signal-to-interference and noise ratio for user n on tone k
t	Time index
t_{max}	Maximum number of fully canceled tones for an entire binder with S-TS
$t_{m,k}$	Number of receivers who do not want to suffer from crosstalk produced by transmitter m on their kth tone
t_n	Number of tones currently fully canceled for line n in S-TS
T	DMT symbol duration
$T_{\text{superframe}}$	Duration of a superframe
V_L	Voltage across load impedance Z_L
V_{no}	Voltage across load impedance without transmission line

V_S	Source voltage
$W_k^{(n,m)}$	Element of canceler filter matrix $\mathbf{W}_k$
$W_{QoS,k}^{(n,m)}$	Element of partial canceler filter matrix $\mathbf{W}_{QoS,k}$ when data rate constraints are considered
$\bar{\mathbf{w}}_k^n$	Reduced crosstalk cancellation filter vector for user n on tone k
$\bar{\mathbf{w}}_{QoS,k}^n$	Reduced canceler filter vector for user m on tone k when data rate constraints are considered
$\mathbf{W}_k$	Partial cancellation filter matrix on tone k
$\mathbf{W}_{QoS,k}$	Partial cancellation filter matrix on tone k when data rate requirements are considered
$x_k^{(n)}$	Frequency-domain transmit signal of line n on tone k
$\hat{x}_k^{(n)}$	Estimated frequency-domain transmit signal for user n on tone k
$\tilde{x}_k^{(n)}$	Precoded frequency-domain transmit signal of user n on tone k
x_i	DMT transmit symbol sample
X_{dB}	Amplitude offset of the crosstalk transfer function in dB
X_k	Transmit modulation symbol on the kth tone
$\hat{X}_k$	Estimated modulation symbol on the kth tone
$\mathbf{x}_k$	Transmit signal vector on tone k
$\bar{\mathbf{x}}_k^n$	Reduced frequency-domain transmit signal vector of user n on tone k for precoding
$\hat{\mathbf{x}}_k$	Estimated frequency-domain received signal vector on tone k
$\tilde{\mathbf{x}}_k$	Precoded frequency-domain transmit signal vector on tone k
$y_k^{(n)}$	Frequency-domain receive signal of line n on tone k
y_i	DMT receive symbol sample

Y_k	Received modulation symbol on the kth tone
$\mathbf{y}_k$	Receive signal vector on tone k
$\bar{\mathbf{y}}_k^n$	Reduced frequency-domain receive signal vector for user n on tone k
$Z_k^{(n,m)}$	Element of precoder matrix $\mathbf{P}_k$
Z_0	Characteristic impedance
Z_L	Load impedance
Z_S	Source impedance
$\bar{\mathbf{z}}_k^m$	Reduced precoding filter vector for user m on tone k
$\bar{\mathbf{z}}_{QoS,k}^m$	Reduced precoding filter vector for user m on tone k when data rate constraints are considered
$\mathbf{Z}_k$	Precoder matrix
$\mathbf{Z}_{QoS,k}$	Partial precoder filter matrix for tone k when data rate constraints are considered
Δ	Complexity parameter
Δc	Number of crosstalkers remaining for cancellation in S-LS after all data rate targets are achieved
Δf	Tone spacing
Δp	Number of crossalker-tone pairs remaining for cancellation in S-JTLS after all data rate targets are achieved
Δt	Number of tones remaining for full cancellation in S-TS after all data rate targets are achieved
$\gamma(f)$	Propagation constant at frequency f
γ_c	Coding gain
γ_m	Noise margin

γ_{max}	Maximum margin
Γ	SNR gap to capacity
$\Gamma(P_e)$	Shannon gap
κ_{FEXT}	Coupling factor
λ	Lagrange multiplier
μ	Waterfilling level
μ_{pa}	Waterfilling level for power-adaptive waterfilling
μ_{ra}	Waterfilling level for rate-adaptive waterfilling
ν	Length of cyclic prefix
$\phi(f)$	Phase of crosstalk transfer function
$\phi_{H_k^{(n,m)}}$	Phase of crosstalk channel transfer factor $H_k^{(n,m)}$
σ^2	Noise variance
$\tilde{\sigma}_n^2(k)$	Noise variance after crosstalk cancellation of user n on tone k
$\sigma_{n,total}^2(k)$	Total noise contribution of user n on tone k
σ_S^2	Transmit signal variance

Bibliography

[ACA$^+$07] Assia, Conexant, Actelis, Ikanos and Upzide. Proposed Definition of the Normalized Error Sample. Technical report, NIPP-NAI Contribution 2007-128, 2007.

[BEP04] H. Bagheri, H. Emami and M. R. Pakravan. *Iterative Joint Power Control and Partial Crosstalk Cancellation in Upstream VDSL*. International Journal on Signal Processing, 1(3): 195–198, August 2004.

[Bha99] V. K. Bhagavath. *Emerging High-Speed xDSL Access Services: Architectures, Issues, Insights, and Implications*. IEEE Communications Magazine, 37(11): 106–114, November 1999.

[BP79] C. A. Belfiore and J. H. Park. *Decision Feedback Equalization*. Proc. of the IEEE, 67(8): 1143–1156, August 1979.

[Bri98] R. F. M. van den Brink. Cable Reference Models for Simulating Metallic Access Networks. Technical report, ETSI STC TM6, 1998.

[Cam98] J. Campello. Optimal Discrete Bit Loading for Multicarrier Modulation Systems. In *Proc. of IEEE International Symposium on Information Theory, Cambridge, MA, USA*, August 1998.

[Cam99] J. Campello. Practical Bit Loading for DMT. In *Proc. of IEEE International Conference on Communications (ICC '99), Vancouver, BC, Canada*, June 1999.

[CCB95] P. S. Chow, J. M. Cioffi and J. A. C. Bingham. *A Practical Discrete Multitone Transceiver Loading Algorithm for Data Transmission over Spectrally Shaped Channels*. IEEE Transactions on Communications, 43(2/3/4): 773–775, Feb./March/April 1995.

[Cen04] R. Cendrillon. *Multi-User Signal and Spectra Co-ordination for Digital Subscriber Lines*. Phd thesis, Katholieke Universiteit Leuven, 2004.

[CFG02] J. M. Cioffi, J. L. Fang and G. Ginis. MIMO Model for Copper Cable: Quantitative Analysis of Matching to Measured Data. Technical report, IEEE 802.3 Ethernet in the First Mile Task Force, 2002.

[CGBM06] R. Cendrillon, G. Ginis, E. van den Bogaert and M. Moonen. *A Near-Optimal Linear Crosstalk Canceler for Upstream VDSL*. IEEE Transactions on Signal Processing, 54(8): 3136–3146, August 2006.

[CGBM07] R. Cendrillon, G. Ginis, E. van den Bogaert and M. Moonen. *A Near-Optimal Linear Crosstalk Precoder for Downstream VDSL*. IEEE Transactions on Communications, 55(5): 860–863, May 2007.

[CGMA04] R. Cendrillon, G. Ginis, M. Moonen and K. van Acker. *Partial Crosstalk Precompensation in Downstream VDSL*. Signal Processing, 84(11): 2005–2019, November 2004.

[CHCM07] R. Cendrillon, Jianwei Huang, Mung Chiang and M. Moonen. *Autonomous Spectrum Balancing for Digital Subscriber Line*. IEEE Transactions on Signal Processing, 55(8): 4241–4257, March 2007.

[Cio91] J. M. Cioffi. A Multicarrier Primer. Technical report, Stanford University/ Amati T1E1 contribution, I1E1.4/91-157, 1991.

[CKLC03] Seong Taek Chung, Seung Jean Kim, Jungwon Lee and J. M. Cioffi. A Game-Theoretic Approach to Power Allocation in Frequency-Selective Gaussian Interference Channels. In *Proc. of IEEE International Symposium on Information Theory*, June/July 2003.

[CM05] R. Cendrillon and M. Moonen. Iterative Spectrum Balancing for Digital Subscriber Lines. In *Proc. of IEEE International Conference on Communications (ICC '05), Seoul, Korea*, May 2005.

[CMA$^+$03] R. Cendrillon, M. Moonen, K. van Acker, T. Bostoen and P. Vandaele. Partial Crosstalk Cancellation Exploiting Line and Tone Selection in Upstream VDSL. In *Proc. of Baiona Workshop on Signal Processing in Communications, Baiona, Spain*, September 2003.

[CMG+04] R. Cendrillon, M. Moonen, G. Ginis, K. van Acker, T. Bostoen and P. Vandaele. *Partial Crosstalk Cancellation for Upstream VDSL*. EURASIP Journal on Applied Signal Processing, 2004(10): 1520–1535, August 2004.

[CMV+04a] R. Cendrillon, M. Moonen, J. Verlinden, T. Bostoen and G. Ginis. Improved Linear Crosstalk Precompensation for DSL. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP '04), Montreal, QC, Canada*, 2004.

[CMV+04b] R. Cendrillon, M. Moonen, J. Verlinden, T. Bostoen and Wei Yu. Optimal Multiuser Spectrum Management for Digital Subscriber Lines. In *Proc. of IEEE International Conference on Communications (ICC '04), Paris, France*, June 2004.

[CYM+06] R. Cendrillon, Wei Yu, M. Moonen, J. Verlinden and T. Bostoen. *Optimal Multiuser Spectrum Balancing for Digital Subscriber Lines*. IEEE Transactions on Communications, 54(5): 922–933, May 2006.

[DBM+08] P. Duvaut, P. Biyani, A. Mahadevan, S. Singh and S. Maheshwari. Adaptive Off-Diagonal MIMO Canceller (ODMC) for VDSL Upstream Self FEXT Mitigation. In *Proc. of 16th European Signal Processing Conference (EUSIPCO '08), Lausanne, Switzerland*, August 2008.

[DMR12] M. Düngen, A. G. Monsalve and H. Rohling. *Quality-of-Service-Based Interference Cancellation in Upstream Very-High-Bit-Rate Digital Subscriber Line Systems*. IET Communications, 6(11): 1382–1387, July 2012.

[DMS+07] P. Duvaut, A. Mahadevan, M. Sorbara, E. Langberg and P. Biyani. Adaptive Off-Diagonal MIMO Precoder (ODMP) for Downstream DSL Self FEXT Cancellation. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM '07), Washington DC, USA*, November 2007.

[Dün12] M. Düngen. *Joint Partial Crosstalk Cancellation and Spectrum Management for xDSL Systems with Data Rate Constraints*. Frequenz, 66(5-6): 129–134, May 2012.

[DP02] H. Dai and H. V. Poor. *Turbo Multiuser Detection for Coded DMT VDSL Systems*. IEEE Journal on Selected Areas in Communications, 20(2): 351–362, February 2002.

[DR10] M. Düngen and H. Rohling. Channel Adaptation in VDSL2 Systems. In *Proc. of 15th International OFDM-Workshop (InOWo '10), Hamburg, Germany*, September 2010.

[DRR10] M. Düngen, Y. Ruan and H. Rohling. Crosstalk Cancellation in VDSL Systems. In *Proc. of 18th European Signal Processing Conference (EUSIPCO), Aalborg, Denmark*, August 2010.

[FH96] R. F. H. Fischer and J. B. Huber. A New Loading Algorithm for Discrete Multitone Transmission. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM '96), London, UK*, November 1996.

[FJK+02] C. Faller, Biing-Hwang Juang, P. Kroon, Hui-Ling Lou, S.A. Ramprashad and C.-E. W. Sundberg. *Technical Advances in Digital Audio Radio Broadcasting*. Proc. of the IEEE, 90(8): 1303–1333, August 2002.

[GA04] Y. George and O. Amrani. Bit Loading Algorithms for OFDM. In *Proc. of International Symposium on Information Theory (ISIT '04), Chicago, IL, USA*, June/July 2004.

[GC00] G. Ginis and J. M. Cioffi. A Multi-User Precoding Scheme Achieving Crosstalk Cancellation with Application to DSL Systems. In *Proc. of 34th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA*, October/November 2000.

[GC01] G. Ginis and J. M. Cioffi. Vectored-DMT: A FEXT Canceling Modulation Scheme for Coordinating Users. In *Proc. of IEEE International Conference on Communications (ICC '01), Helsinki, Finland*, May 2001.

[GDJ06] P. Golden, H. Dedieu and K. S. Jacobsen. *Fundamentals of DSL Technology*. Auerbach Publications, 1st edition, 2006.

[GDJ08] P. Golden, H. Dedieu and K. S. Jacobsen. *Implementations and Applications of DSL Technology*. Auerbach Publications, 1st edition, 2008.

[HM72] H. Harashima and H. Miyakawa. *Matched-Transmission Technique for Channels With Intersymbol Interference*. IEEE Transactions on Communications, 20(4): 774–780, August 1972.

[Jac01] K. S. Jacobsen. *Methods of Upstream Power Back-Off on Very High Speed Digital Subscriber Lines*. IEEE Communications Magazine, 39(3): 210–216, March 2001.

[LCJ+07] Bin Lee, J. M. Cioffi, S. Jagannathan, Kibeom Seong, Youngjae Kim, M. Mohseni and M. H. Brady. *Binder MIMO Channels*. IEEE Transactions on Communications, 55(8): 1617–1628, August 2007.

[Lev01] H. E. Levin. A Complete and Optimal Data Allocation Method for Practical Discrete Multitone Systems. In *Proc. of IEEEGlobal Telecommunications Conference (GLOBECOM '01), San Antonio, TX, USA*, November 2001.

[Lük92] H. D. Lüke. *Korrelationssignale*. Springer Verlag, 1st edition, 1992.

[LKBC06] W. Lee, Youngjae Kim, M. H. Brady and J. M. Cioffi. SPC09-6: Band-Preference Dynamic Spectrum Management in a DSL Environment. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM '06), San Francisco, CA, USA*, November/December 2006.

[LMOW07] J. Le Masson, M. Ouzzif and I. Wahibi. Channel Estimation Using Data and Pilots for a Coordinated DSL System. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM '07), Washington DC, USA*, November 2007.

[LP06] Zhi-Quan Luo and Jong-Shi Pang. *Analysis of Iterative Waterfilling Algorithm for Multiuser Power Control*. EURASIP Journal on Advances in Signal Processing, 2006(024012): 1–10, March 2006.

[LV06a] J. Louveaux and A.-J. van der Veen. *Adaptive DSL Crosstalk Precancellation Design Using Low-Rate Feedback From End Users*. IEEE Signal Processing Letters, 13(11): 665–668, November 2006.

[LV06b] J. Louveaux and A.-J. van der Veen. *Error Sign Feedback as an Alternative to Pilots for Tracking of FEXT Transfer Functions in Downstream VDSL*. EURASIP Journal on Applied Signal Processing, 2006(94105): 1–14, March 2006.

[LY05] R. Lui and Wei Yu. Low-Complexity Near-Optimal Spectrum Balancing for Digital Subscriber Lines. In *Proc. of IEEE International Conference on Communications (ICC '05), Seoul, Korea*, May 2005.

[MGP09] J. Maes, M. Guenach and M. Peeters. Statistical Channel Model for Gain Quantification of DSL Crosstalk Mitigation Techniques. In *Proc. of IEEE International Conference on Communications (ICC '09), Dresden, Germany*, June 2009.

[MMS09] M. Maesoumi and M. A. Masnadi-Shirazi. *Joint Partial Crosstalk Cancellation and Modified Iterative Water-Filling for Upstream VDSL*. AEU - International Journal of Electronics and Communications, 63(10): 879–888, October 2009.

[PE09] J. Papandriopoulos and J. S. Evans. *SCALE: A Low-Complexity Distributed Protocol for Spectrum Balancing in Multiuser DSL Networks*. IEEE Transactions on Information Theory, 55(8): 3711–3724, August 2009.

[PR80] A. Peled and A. Ruiz. Frequency Domain Data Transmission Using Reduced Computational Complexity Algorithms. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '80), Denver, CO, USA*, April 1980.

[Pro01] J. G. Proakis. *Digital Communications*. McGraw-Hill International Edition, 4th edition, 2001.

[Ran08] J. Ran. *Signal Processing, Channel Estimation and Link Adaptation in MIMO-OFDM Systems*. Phd thesis, TU Hamburg-Harburg, 2008.

[RCK92] A. Ruiz, J. M. Cioffi and S. Kasturia. *Discrete Multiple Tone Modulation with Coset Coding for the Spectrally Shaped Channel*. IEEE Transactions on Communications, 40(6): 1012–1029, June 1992.

[RDR08] Y. Ruan, M. Düngen and H. Rohling. Crosstalk Cancellation in VDSL Systems. In *Proc. of 13th International OFDM Workshop (InOWo '08), Hamburg, Germany*, August 2008.

[RDR09] Y. Ruan, M. Düngen and H. Rohling. QoS-Based Partial Crosstalk Cancellation in Upstream VDSL Systems. In *Proc. of 14th International OFDM Workshop (InOWo '09), Hamburg, Germany*, September 2009.

[RDR11] Y. Ruan, M. Düngen and H. Rohling. *Successive Crosstalk Cancellation Facing the Dynamic Situation in Very-High-Speed Digital Subscriber Line 2 Systems*. IET Communications, 5(11): 1491–1496, July 2011.

[Rei04] U. Reimers. *DVB: The Family of International Standards for Digital Video Broadcasting*. Springer-Verlag, 2nd edition, 2004.

[RF09] H. Rohling and C. Fellenberg. *Successive bit loading scheme*. Electronics Letters, 45(4): 214–216, February 2009.

[Roh11] H. Rohling. *OFDM: Concepts for Future Communication Systems*. Springer-Verlag, 1st edition, 2011.

[Rua14] Y. Ruan. *Interference Cancellation Techniques for Multiple-Line Transmission in Modern DSL Systems*. Phd thesis, TU Hamburg-Harburg, 2014.

[SCGC02] Kee Bong Song, Seong Taek Chung, G. Ginis and J. M. Cioffi. *Dynamic Spectrum Management for Next-Generation DSL Systems*. IEEE Communications Magazine, 40(10): 101–109, October 2002.

[Sch02] S. Schelstraete. *Defining Upstream Power Back-Off for VDSL*. IEEE Journal on Selected Areas in Communications, 20(5): 1064–1074, June 2002.

[SCS99] T. Starr, J. M. Cioffi and P. J. Silverman. *Understanding Digital Subscriber Line Technology*. Prentice Hall PTR, 1st edition, 1999.

[SDS$^+$07] M. Sorbara, P. Duvaut, F. Shmulyian, S. Singh and A. Mahadevan. Construction of a DSL-MIMO Channel Model for Evaluation of FEXT Cancellation Systems in VDSL2. In *Proc. of 2007 IEEE Sarnoff Symposium, Princeton, NJ, USA*, April/May 2007.

[Sjö00] F. Sjöberg. A VDSL Tutorial. Technical report, Luleå Tekniska Universitet, available at http://epubl.luth.se/1402-1528/2000/02/LTU-FR-0002-SE.pdf, 2000.

[SSCS03] T. Starr, M. Sorbara, J. M. Cioffi and P. J. Silverman. *DSL Advances*. Prentice Hall, 1st edition, 2003.

[Sta11] Very High Speed Digital Subscriber Line Transceivers 2 (VDSL2). Technical report, ITU-T Std. G.993.2, 2011.

[TH00] G. Tauböck and W. Henkel. MIMO Systems in the Subscriber-Line Network. In *Proc. of 5th International OFDM-Workshop (InOWo '00), Hamburg, Germany*, September 2000.

[Tom71] M. Tomlinson. *New Automatic Equaliser Employing Modulo Arithmetic*. IET Electronics Letters, 7(5): 138–139, March 1971.

[TV05] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 1st edition, 2005.

[VBY05] J. Verlinden, T. Bostoen and G. Ysebaert. *Dynamic Spectrum Management for Digital Subscriber Lines - Edition 2*. Technology White Paper - Alcatel, June 2005.

[VMGP10] J. Vangorp, M. Moonen, M. Guenach and M. Peeters. *Downstream Power Back-Off in CO/RT-Deployed xDSL Networks*. IEEE Transactions on Communications, 58(2): 453–456, February 2010.

[VTM+06] J. Vangorp, P. Tsiaflakis, M. Moonen, J. Verlinden and K. van Acker. Optimal Spectrum Balancing in Multi-User xDSL Systems With On/Off Power Loading. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06), Toulouse, France*, May 2006.

[WE71] S. B. Weinstein and P. M. Ebert. *Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform*. IEEE Transactions on Communication Technology, 19(5): 628–634, October 1971.

[Wer91] J.-J. Werner. *The HDSL Environment*. IEEE Journal on Selected Areas in Communications, 9(6): 785–800, August 1991.

[WS91] W. Walkoe and T. J. J. Starr. *High Bit Rate Digital Subscriber Line: A Copper Bridge to the Network of the Future*. IEEE Journal on Selected Areas in Communications, 9(6): 765–768, August 1991.

[XSH09] W. Xu, C. Schroeder and P. A. Hoeher. A Stochastic MIMO Model for Far-End Crosstalk in VDSL Cable Binders. In *Proc. of IEEE International Conference on Communications (ICC '09), Dresden, Germany*, June 2009.

[YC00] Wei Yu and J. M. Cioffi. Multiuser Detection for Vector Multiple Access Channels Using Generalized Decision Feedback Equalization. In *Proc. of 5th International Conference on Signal Processing (WCCC-ICSP '00), Beijing, China*, August 2000.

[YGC02] Wei Yu, G. Ginis and J. M. Cioffi. *Distributed Multiuser Power Control*. IEEE Journal on Selected Areas in Communications, 20(5): 1105–1115, June 2002.

[Yu02] Wei Yu. *Competition and Coordination in Multi-User Communication Environments*. Phd thesis, Stanford University, 2002.

Curriculum Vitae

Family name	Düngen
First name	Monique
Date of birth	April 22nd, 1982
Place of birth	Moers, Germany

08.1987 – 03.1991	Elementary school Obermeitingen, Germany
03.1991 – 06.1992	Elementary school Diez, Germany
08.1992 – 06.2001	Sophie-Hedwig Gymnasium Diez, Germany Degree: 'Abitur' certificate
10.2001 – 04.2007	Studies of Electrical Engineering at Hamburg University of Technology, Hamburg, Germany Degree: Diploma
06.2007 – 09.2011	Scientific staff at Hamburg University of Technology, Hamburg, Germany
02.2012 – today	Development engineer at Robert Bosch GmbH, Hildesheim, Germany