A Flexible and Real-Time Constrained Controller for Sparse Linear Zero-Forcing Based DSL Vectoring

Paschalis Tsiaflakis and Marc Moonen
KU Leuven - Dept. ESAT-SCD / iMinds - Future Health Department, B-3001 Heverlee-Leuven, Belgium
e-mail:paschalis.tsiaflakis@esat.kuleuven.be, marc.moonen@esat.kuleuven.be

***Abstract*—Vectoring is recognized as a very efficient technique to tackle crosstalk interference in multi-user DSL systems. In this paper we focus on sparse linear zero-forcing based vectoring schemes with a shared resource pool, which are known to obtain a good trade-off between online computational complexity and near-optimal performance. We propose a new system setup with a controller that dynamically (in time) optimizes and allocates computational complexity resources taking into account variable data rate demands with real-time constraints. The controller is designed to minimize the online computational complexity over time and allows for flexible fixed and variable data rate allocations. Simulation results with variable real-time constrained video streams for a realistic downstream VDSL scenario demonstrate that this novel controller-based approach allows to significantly reduce the average as well as peak online computational complexity.**

I. Introduction

Digital subscriber line (DSL) technology enables high-speed internet access over the local telephone network. It provides a cost-efficient broadband access solution for residential as well as business users. However, one major impairment in current DSL networks is the presence of electromagnetic coupling among different DSL lines within the same cable bundle. This leads to interference between the users' data transmission, also referred to as *crosstalk*. Crosstalk transforms the local DSL network into a challenging interference-limited multi-user environment. To prevent significant reductions in achievable data rates, multi-user coordination techniques for crosstalk mitigation have been proposed in the literature and are already being applied in practice.

One efficient multi-user coordination technique is *vectoring*. This consists of the joint processing of the DSL signals at the (network side) access node, leading to coordinated upstream reception or coordinated downstream transmission. Linear zero-forcing (ZF) based schemes are recognized as very efficient vectoring schemes as they require a very low online computational complexity (compared to non-linear schemes such as the generalized decision feedback canceller), have a low-latency design and display near-optimal performance because of the diagonally dominant (row-wise or column-wise) structure of DSL channels [1][2]. To further reduce the computational complexity of linear ZF based schemes, partial crosstalk cancellation schemes have been proposed in [1][2][3][4]. These *sparse linear ZF based schemes* resort to cancelling only part of the crosstalk, and display a good trade-off between further reducing the computational complexity compared to (full) linear ZF based schemes and maintaining near-optimal data rate performance. For instance, it has been shown in [1][2] that for a computational complexity budget of 20/30% a near-optimal data rate performance of 80/90% can be achieved for downstream/upstream sparse linear crosstalk precoding/cancellation, respectively. In this paper, we focus on such DSL vectoring with sparse linear ZF based schemes where a given computational complexity budget is shared amongst multiple DSL users, which can be seen as a *shared resource pool*. Our aim is to improve this system by introducing a *controller* that *dynamically (in time)* optimizes and allocates computational complexity resources to the users, depending on their fixed as well as variable demands.

Current DSL service providers offer a range of DSL subscriptions, each of them characterized by a specific down- and upstream data rate. However, these offered data rates are mostly static, i.e., the same data rates are allocated at all time regardless of whether the broadband connections are used or not. Given that a typical end-user is not active during all of the day, this static allocation is not aligned with the true demand and behaviour of the end-users. A better aligned system setup would consist of *dynamic access* where the end-user only receives bandwidth or only has to pay when he actually uses his broadband connection.

In our considered system model of sparse linear ZF based vectoring with a *shared computational complexity budget*, we propose and investigate a new system setup, in which there are fixed resources for each user that are not necessarily constant over time, and variable resources (from a shared resource pool) that can be allocated to the users *on request/demand*, so as to increase their data rates when needed. The fixed resources guarantee minimum data rates to the end-users at all time to support a minimum service (e.g., for e-mail or skype calling), whereas the variable resources result in temporarily allocated

P. Tsiaflakis is a postdoctoral fellow funded by the Research Foundation-Flanders (FWO). This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Research Council: PFV/10/002 (OPTEC), Bilateral Scientific Cooperation between Tsinghua University & KU Leuven 2012-2014, FWO project G091213N 'Cross-layer optimization with real-time adaptive dynamic spectrum management for fourth generation broadband access networks', Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office: IUAP 'Belgian network on Stochastic modelling, analysis, design and optimization of communication systems' (BESTCOM) 2012-2017, and Concerted Research Action GOA-MaNet. The scientific responsibility is assumed by its authors.

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

additional data rates to support 'occasional' services such as video on demand or video conferencing. This new system setup allows to offer much more *flexibility* to the end-users in their subscriptions. For instance, residential users may only want to be active during weekday evenings and weekends, next to requiring occasional additional bandwidth to support, for instance, a movie download, whereas business users may want to be active in the daytime with only a minimal (elastic) service during nights and weekends.

Working with a shared resource pool of computational complexity that can be dynamically allocated to the users, offers the service provider the possibility to obtain a significant statistical multiplexing gain, as users can be allocated temporarily a much larger proportion of resources, i.e., data rates, compared to the (worst-case) fixed allocation of current practice. This is expected to better accommodate peak demands.

Given the envisaged dynamic setting, we also introduce the feature of *real-time stream*[1] *constraints over time*. This means that the end-user cannot only reserve resources (services and data) on-demand, but can also specify an end-time for the delivery of the services or data. For instance, the user can specificy the exact (future) play-time of a movie in advance. We will show that this allows the network service provider to schedule the data access optimally over time so as to minimize the allocated resources, which corresponds to a *reservation-based resource scheduling* approach. Requests for additional variable resources come at a certain price for the end-user, converging to a 'pay per demand' setting. The price can be made dependent on the requested delivery time or delay, i.e., the time from request (start-time) until end-time, where a later delivery is less expensive, as it causes a smaller load to the system[2].

The goal of this paper is to propose an optimization based controller design for the new dynamic system setup and to assess the potential benefit in terms of flexibility and computational complexity efficiency of the resulting DSL system.

Sparse linear ZF based vectoring schemes have been presented in [1][2][3][4]. These works however consider a static resource allocation approach. Research work on transitioning from a static to a dynamic allocation of vectoring resources has been presented in [5][6][7]. These works however only consider the past user behaviour in terms of accumulated queue loads, whereas in this paper future user demands and flexible fixed data rates are considered. Research work that exploits the reservation of resources for real-time delivery of video on demand has been presented in [8][9][10].

II. Resource System Model

We consider a cable bundle that consists of a set $\mathcal{N} = \{1, \ldots, N\}$ of DSL users with the standard assumption of synchronous discrete multi-tone (DMT) modulation with a set $\mathcal{K} = \{1, \ldots, K\}$ of tones (i.e., frequency carriers). Considering linear ZF based schemes and taking into account the (row-wise or column-wise) diagonally dominant structure of DSL channels, the bit loading of user n on tone k and in time slot t can be characterized as follows [1][2],

$$b_k^{n,t}(\mathbf{c}_k^{n,\cdot,t}) = \log_2\left(1 + \frac{s_k^n}{\sum_{m\in\mathcal{N}\setminus n}(1 - c_k^{n,m,t})h_k^{n,m}s_k^m + z_k^n}\right),$$

where s_k^n denotes the transmit power of user n on tone k, z_k^n denotes the noise power of user n on tone k, and $h_k^{n,m}$ denotes the channel gain on tone k from transmitting user m to receiving user n. Both z_k^n and $h_k^{n,m}$ are normalized with the direct channel and the standard SNR gap. Note that we take a discrete time-slotted approach where the time slots are indexed with t, which is in contrast to the typical static approach in most of the multi-user DSL transmission literature, e.g., [1][2][3][4]. $c_k^{n,m,t}$ denotes a cancellation tap of the linear ZF based vectoring and takes a value of 1 if user n cancels the crosstalk contribution from user m on tone k during time slot t, and 0 otherwise. $\mathbf{c}_k^{n,\cdot,t}$ denotes a vector consisting of $N-1$ elements $c_k^{n,m,t}$ with $m \in \mathcal{N}\setminus n$.

As a result the user's total achievable data rate in time slot t, i.e., $R^{n,t}$, is in function of the cancellation taps and can be expressed as follows,

$$\begin{aligned} R^{n,t} &\triangleq R^{n,t}(\mathbf{c}_\cdot^{n,\cdot,t}) = \sum_{k\in\mathcal{K}} b_k^{n,t}(\mathbf{c}_k^{n,\cdot,t}) \\ R^{n,t} &\geq R_{\min}^{n,t} = \sum_{k\in\mathcal{K}} b_k^{n,t}(\mathbf{0}_{N-1}) \\ R^{n,t} &\leq R_{\max}^{n,t} = \sum_{k\in\mathcal{K}} b_k^{n,t}(\mathbf{1}_{N-1}), \end{aligned}$$

where $R_{\min}^{n,t}$ and $R_{\max}^{n,t}$ are constants denoting the minimum and maximum achievable data rate for user n in time slot t, respectively, and where $\mathbf{0}_{N-1}$ and $\mathbf{1}_{N-1}$ denote the all-zero and all-one unit vector, respectively, with $N-1$ elements.

Finally there is a total cancellation tap constraint per time slot as follows,

$$\sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{N}\setminus n} c_k^{n,m,t} \leq \alpha NK(N-1), \qquad (1)$$

where α is a number between 0 and 1 to indicate the available proportion of cancellation taps compared to full (non-sparse) vectoring. The cancellation taps are thus the computational complexity resources in our system model and they can be allocated to the users according to their demands. Constraint (1) couples the allocation of the resources over the users, and constrains the number of available resources. As such, it forms a *shared resource pool*.

III. Controller for Sparse Linear ZF Based DSL Vectoring

Given a shared resource pool of cancellation tap resources and constrained by (1), the aim is to allocate these over the users so as to cope with their fixed and variable demands.

We model the fixed user demands (related to the users' subscription) by fixed minimum data rate constraints $R_{\text{Fix}}^{n,t}$. Note that these are constants (as they are known in advance)

[1]Stream and demand are used interchangeably throughout the text.

[2]It is beyond the scope of this paper to address the impact of pricing/charging schemes on the considered system setup.

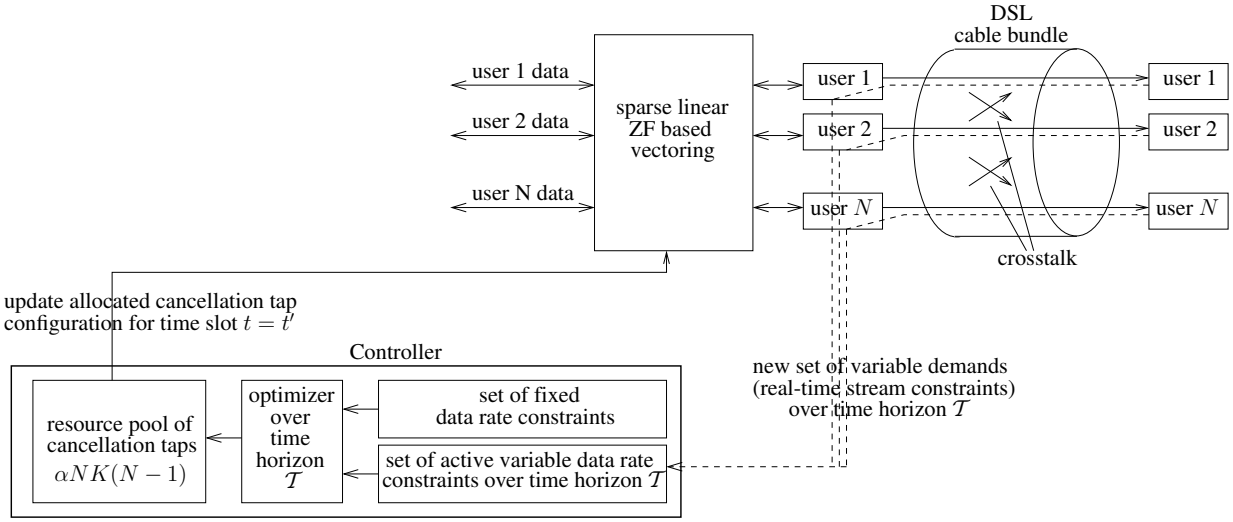


Fig. 1. Controller-based system for sparse linear ZF-based DSL vectoring with a shared resource pool of computational complexity resources (cancellation taps).

but are not necessary constant over time. The variable demands are however not known in advance but only at the time that a user makes an additional request. Moreover, the variable demands involve a start-time of the service request as well as an end-time, as was already highlighted in Section I. Whenever a new user demand arrives, which may require additional bandwidth, we model it as a real-time stream constraint over time (including start- and end-time). This requires to extend the traditional static approach of DSL vectoring where only a one-time configuration is considered to a time-dynamic approach, where the allocation is considered over a time slot horizon $\mathcal{T} = \{t', \ldots, t' + T - 1\}$ with length T. For this, we introduce a controller that (at each time slot) computes the optimal resource allocation for the whole time slot horizon $\mathcal{T}$, given the current known set of fixed and variable demands. However, the optimized resource allocation of the sparse linear ZF based vectoring is only applied in the current time slot $t = t'$. This is also visualized in Figure 1. The fixed data rate constraints are known in advance in the controller, and the set of variable demands are updated every time a new request arrives or a queued request is (partly or fully) processed. We want to highlight that the true duration of the time slots is irrelevant from a system concept point of view, and one may consider short time slots of a few milliseconds as well as time slots of several hours. The time slot duration only defines the time available to the controller to solve the corresponding optimization problem, and to perform the hardware configurations of the new resource allocation in the sparse linear ZF-based vectoring.

To model the above controller and to take the underlying vectoring scheme into account, we first introduce two additional variables $R_{\mathrm{RT}}^{n,f_n,t}$ and $R_{\mathrm{EX}}^{n,f_n,t}$ which refer to the variable data rate allocated to stream $f_n \in \mathcal{F}_n$ of user n in time slot t, and the excess data rate for stream $f_n \in \mathcal{F}_n$ of user n in time slot t, respectively. $\mathcal{F}_n$ refers to the active set of streams for user n. There can be multiple streams per user, i.e., $\mathrm{cardinality}(\mathcal{F}_n) \nleq 1$. The excess data rate is the data rate that is available for processing demands for user n whenever $R_{\min}^{n,t}$ is larger than $R_{\mathrm{FIX}}^{n,t}$. This excess data rate does not require the allocation of cancellation taps and is wasted if there are no active demands to be processed for user n. The variable data rate $R_{\mathrm{RT}}^{n,f_n,t}$ is the additional data rate to process the demand for user n, but requires the allocation of cancellation taps so that a data rate larger than $R_{\min}^{n,t}$ can be achieved. To minimize the computational complexity of the whole system, we propose the following optimization problem for the controller,

$$\min \quad \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{N}\setminus n} c_k^{m,n,t} \tag{2}$$

$$\text{s.t.} \quad c_k^{m,n,t} \in \{0,1\}, \forall t, n, k, m \neq n \tag{3}$$

$$R_{\mathrm{RT}}^{n,f_n,t} \geq 0, \forall t, n, f_n \tag{4}$$

$$R_{\mathrm{EX}}^{n,f_n,t} \geq 0, \forall t, n, f_n \tag{5}$$

$$\sum_{f_n\in\mathcal{F}_n} R_{\mathrm{EX}}^{n,f_n,t} = \left[0, R_{\min}^{n,t} - R_{\mathrm{FIX}}^{n,t}\right]_{\mathrm{M}}, \forall n, t \tag{6}$$

$$R^{n,t} \geq \left[R_{\mathrm{FIX}}^{n,t}, R_{\min}^{n,t}\right]_{\mathrm{M}} + \sum_{f_n\in\mathcal{F}_n} R_{\mathrm{RT}}^{n,f_n,t}, \forall n, t \tag{7}$$

$$\sum_{t\in\mathcal{T}_{\{n,f_n\}}} \left(R_{\mathrm{RT}}^{n,f_n,t} + R_{\mathrm{EX}}^{n,f_n,t}\right) \geq \bar{R}_{\mathrm{RT}}^{n,f_n}, \forall n, f_n \tag{8}$$

$$\sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{N}\setminus n} c_k^{n,m,t} \leq \alpha NK(N-1), \forall t, \tag{9}$$

with variables $c_k^{n,m,t}, R_{\mathrm{RT}}^{n,f_n,t}, R_{\mathrm{EX}}^{n,f_n,t}, n \in \mathcal{N}, f_n \in \mathcal{F}_n, k \in \mathcal{K}, m \in \mathcal{N}\setminus n, t \in \mathcal{T}$, and which minimizes the allocated cancellation taps over the considered time slot horizon. Operator $[x, y]_{\mathrm{M}}$ refers to a maximum operator $\max(x, y)$. These maximum operators in (6) and (7) are necessary to prevent allocation of cancellation taps when $R_{\min}^{n,t} \geq R_{\mathrm{FIX}}^{n,t}$. Constraints (8) model the variable data demands with both start-time and end-time, where $\bar{R}_{\mathrm{RT}}^{n,f_n}, \forall n, f_n$, corresponds to the (given) data load for demand $f_n \in \mathcal{F}_n$ of user n, $\mathcal{T}_{\{n,f_n\}}$ refers to the set of time slots from start-time until end-time of the demand f_n of user

n. Problem (2)-(9) needs to be solved by the controller at each time slot t' where a time horizon of length T is considered. Only the cancellation tap configuration of timeslot $t = t'$ is then updated with its corresponding data rates. The variable demand constraints (8) are then also updated after time slot t', i.e., the allocated data rates $R_{\mathrm{RT}}^{n,f_n,t'} + R_{\mathrm{EX}}^{n,f_n,t'}, \forall n, f_n$, are subtracted from $\bar{R}_{\mathrm{RT}}^{n,f_n}, \forall n, f_n$, and removed if the demand data are fully processed. New demands that are within the time slot horizon are also continuously collected and added to the set of demand constraints. This gives an adaptive variable demand constraint set that evolves as data is processed and new services are requested by the users. Constraint (9) is the resource pool constraint at each time slot.

The strength of the above controller is that it optimally schedules the cancellation tap allocation over a time slot horizon, which allows it to make improved choices compared to approaches that only consider the current time slot. The concrete practical impact will be evaluated by simulations in Section IV.

Finally we want to note that problem (2)-(9) corresponds to a mixed integer nonlinear optimization problem. We have developed a customized optimization procedure to tackle this problem. However, the description of this optimization procedure is beyond the scope (and page budget) of this paper.

IV. Simulations

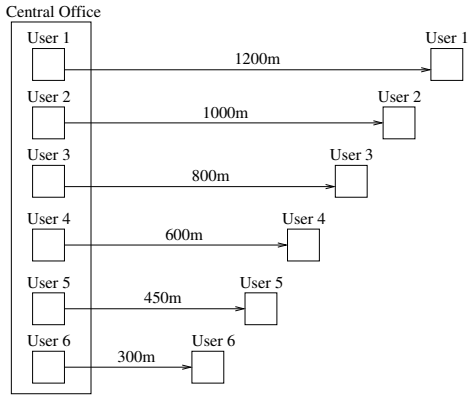


Fig. 2. 6-User downstream VDSL scenario

For our DSL vectoring simulation setup, we consider a cable bundle of 6 interfering downstream (DS) VDSL lines, i.e., $N = 6$ and $K = 1604$, with different line lenghts as depicted in Figure 2. The DSL lines are modelled as 24AWG twisted copper pairs. The SNR gap is chosen to be 12.9 dB, corresponding to a coding gain of 3 dB, a noise margin of 6 dB and a target symbol error probability of $10^{-}7$. The tone spacing is 4.3125 kHz and the DMT symbol rate is 4 kHz. The transmit spectra are fixed at -60 dBm/Hz. Furthermore, the system setup consists of a sparse linear ZF based precoder at the access node, as illustrated in Figure 1, with a shared resource pool of cancellation taps of 30% of a full crosstalk cancellation, i.e., $\alpha = 0.3$ in (1). The cancellation taps can be dynamically allocated by the controller at each time slot.

TABLE I
Variable demand streams with real-time constraints (start- and end-time), corresponding to video on demand (with SD or HDTV format) or video-conferencing.

Stream	User	Start time slot	End time slot	Format
1	4	11	11	SD
2	4	11	23	HD
3	4	11	59	HD
4	6	20	20	4xVC
5	3	29	35	HD
6	1	40	47	SD
7	2	52	59	SD
8	5	57	57	VC
9	4	59	143	SD
10	2	64	71	SD
11	4	70	71	HD
12	4	95	107	HD
13	6	104	104	4xVC
14	3	125	131	HD
15	4	129	129	VC
16	2	136	143	SD
17	5	141	141	VC
18	2	148	155	SD
19	4	179	191	HD
20	3	185	191	HD
21	6	188	188	4xVC
22	1	208	215	SD
23	2	220	227	SD
24	5	225	225	VC
25	4	227	311	HD
26	2	232	239	SD
27	3	257	263	HD
28	4	263	275	HD
29	6	272	272	4xVC
30	2	304	311	SD
31	5	309	309	VC
32	2	316	323	SD
33	3	317	323	HD

We consider time slots of 2 hours, as our intention in this paper is to model the impact and potential of the proposed controller-based scheme for video on demand and video-conferencing applications. Time slot 1 corresponds to the two-hour period 0:00am-2:00am on Monday. We consider the following three types of stream formats with corresponding data rates: (1) 1.5 Mbps Standard Definition TeleVision (SDTV), 9 Mbps High Definition TeleVision (HDTV) and 400 Kbps real-time video conferencing streams. For the time slot horizon, the controller considers a full week in advance at each time slot, i.e., $T = 84$. This means that the end-user can schedule the end-time of a stream up to a week later.

To assess the performance for a DSL system with different types of users, we associate the DSL lines with different subscriptions in terms of the fixed data rates $R_{\min}^{n,t}$. Lines 1 and 2 correspond to residential users that have a DSL subscription with constant fixed DS data rates of 20 Mbps and 30 Mbps, respectively. Line 3 corresponds to a residential user with flexible fixed data rates as follows: 35 Mbps on weekdays from 4pm until 2am and during weekends. Line 4 has a subscription of 35 Mbps at all time. Line 5 and 6 correspond to subscriptions for business users with flexible fixed DS data rates of 38 Mbps during weekdays between 6am until 8pm and no data rates during weekends. Note that these

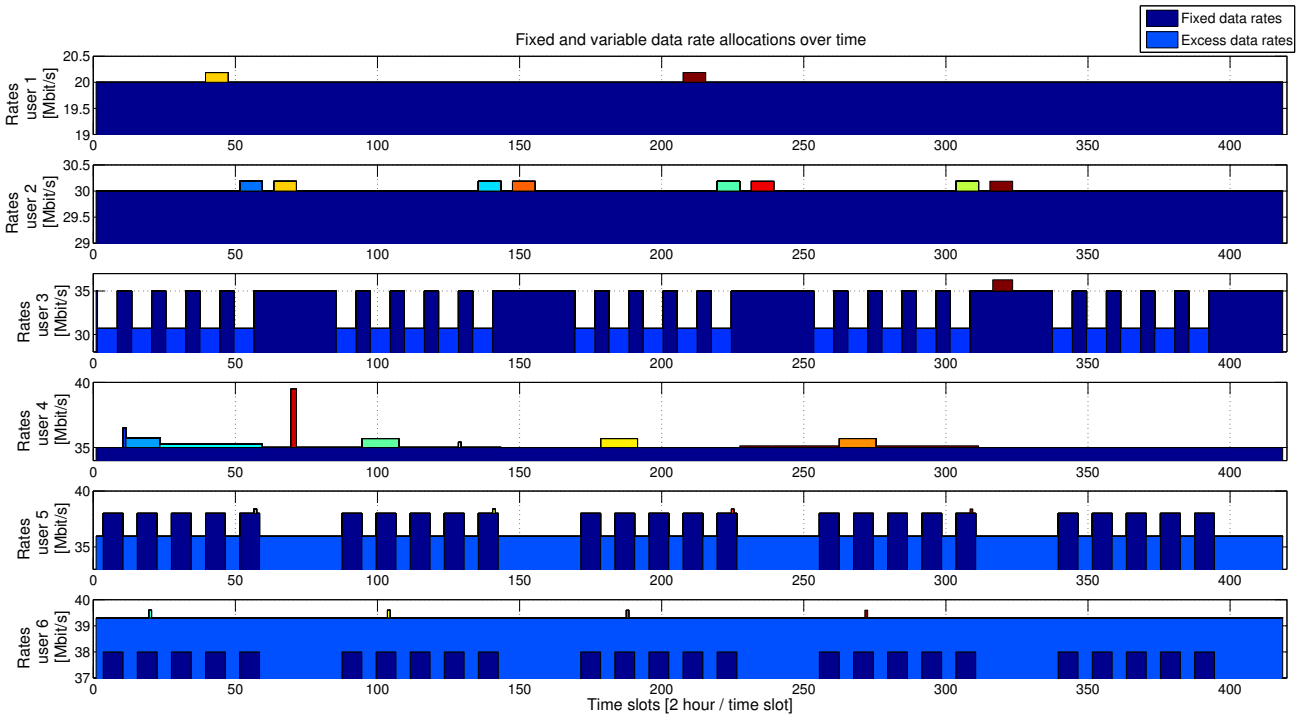


Fig. 3. Fixed and variable data rate allocations over time for downstream VDSL scenario of Figure 2 with fixed data rates schedules in dark blue, excess data rates in light blue, and real time data rates in other colors, satisfying the real time variable stream constraints of Table I.

numbers are merely examples to demonstrate the flexibility of the proposed system setup. Other fixed data rate allocations may also be chosen.

The set of variable demand streams that occur during showtime are summarized in Table I. For instance, stream 6 models a request of user 1 for a video with SD format, which arrives at time slot 40, i.e., on Thursday at 6am, and which has to be completed at time slot 47, i.e., on Thursday at 8pm. Streams 4, 13, 21 and 29, correspond to a request of user 6 for 4 VC streams every Tuesday between 2pm and 4pm. Stream 9 corresponds to a request by user 4 for an SD movie on Friday 8pm to be viewed a week later on Friday 8pm, corresponding to time slots 59 and 143, respectively. Note that user 4 makes frequently use of video on demand requests.

The minimum data rates for user 1 to 6 depend on the line lengths and are, respectively: 17.79 Mbps, 25.23 Mbps, 30.69 Mbps, 33.61 Mbps, 35.95 Mbps and 39.30 Mbps.

For the above outlined system parameter settings, we performed the controller optimization over all time slots covering a total period of five weeks, resulting in the fixed and variable data rate allocations of Figure 3. The dark blue color indicates the fixed data rate allocations, related to the users' subscriptions. The lighter blue color indicates the excess data rates. The other colors indicate the allocation of variable data rates to support the real-time constrained variable streams of Table I, and which require the additional allocation of cancellation taps. The sum of all colors indicates the total data rates allocated to each of the users over the time slots.

A first observation is that the excess data rates are exploited to support (partly or fully) the variable demands, as these require no allocation of cancellation taps, so as to minimize the number of allocated cancellation taps. For instance, it can be seen that user 3 only allocates cancellation taps for stream 33. It does not allocate any cancellation taps for streams 5, 14, 20 and 27, but uses the (already available) excess data rates for this, which are large enough to support these real-time variable streams. Also for user 6, part of the variable demands are supported by the excess data rates and partly by allocated variable data rates (in colors cyan, yellow, red and brown).

A second observation is that the variable data rates for the real-time constrained streams are allocated over the available time (from start-time until end-time) so that a constant data rate level is obtained. For instance, the variable data rates for user 1 are constant between time slot 40 and 47. In fact, the optimal variable data rate allocation follows a *waterfilling* type of allocation of the streams over time. This can also be seen from streams 1, 2, and 3 for user 4. These three streams have the same start-time at time slot 11. The end-time of stream 1 is also time slot 11, meaning that all data of this stream has to be downloaded during time slot 11. As a result the optimization uses all cancellation taps for this particular stream during time slot 11. It is not efficient for streams 2 and 3 to allocate taps during time slot 11. For stream 2 it is more efficient to start from time slot 12 and end at time slot 23. Stream 3 is then covered in time slots 24 up to time slot 59. The optimized waterfilling type of allocation is enabled by the introduction

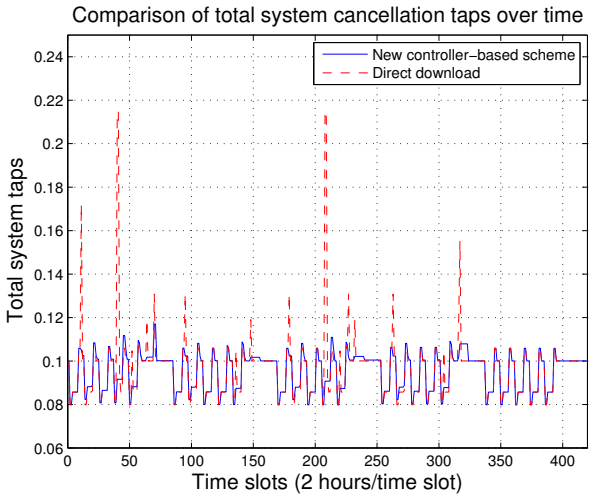


Fig. 4. Total system cancellation tap allocation over time for new controller-based scheme and direct download approach

of the end-time feature. This allows to optimize the data rate allocations over time, with an improvement in computational complexity, i.e., reduced allocated cancellation taps, as a result.

We can now compare our proposed controller-based system with a base-line approach that allocates variable data rates whenever a new request arrives, so as to fully support it during the considered time slot. This is a greedy approach that does not consider an optimization over time, but merely processes the arriving streams as quickly as possible. We will call this the 'direct download' approach. Considering streams 1, 2 and 3, the direct download approach allocates as many cancellation taps as necessary so as to support one SD, and two HD videos during time slot 11. If the data request cannot be supported during one time slot, the remaining part is moved to the next time slot and so on. For our case, the direct download approach results in an excess total tap allocation of 3470 taps to support streams 1, 2 and 3. Our proposed controller, in contrast, only requires 3122 taps, because of the optimization over time. This is an improvement of 10%.

An important advantage of our new controller-based system versus the direct download approach is that the *peak number* of allocated cancellation taps is *significantly smaller*. In Figure 4 we plot the total number of cancellation taps allocated over time for both approaches. It can be seen that the peak of our new controller scheme is only 11.7% versus 21.5% for the direct download scheme, which corresponds to an improvement of 45%. The resource pool size can thus be *dimensioned significantly smaller*.

Finally, we wish to highlight that these dynamic schemes are all much more *computationally efficient* than previously proposed fixed cancellation tap budget approaches [3][4][5], which use a total system budget of 30%. From Figure 4, we can derive that the average number of allocated cancellation taps is only 9.5% for our new scheme, which corresponds to an improvement of 68%.

V. Conclusion

We have proposed a new controller-based system to dynamically control sparse linear ZF-based DSL vectoring with a *shared resource pool* of computational complexity. This system allows to provide *flexible* fixed data rate allocations within an interference-limited DSL network, in addition to variable data rates to support variable user demands with *real-time constraints*, i.e., with start- and end-times. The controller minimizes the (online) system computational complexity, and takes a future time slot horizon into account at each time slot, in contrast to existing dynamic vectoring schemes that only look at past queue loads. This allows for a *reservation-based scheduling* approach. The novel system setup has been evaluated for a realistic downstream VDSL scenario, with variable SD, HDTV and video-conferencing streams. It has been demonstrated that the proposed controller-based system allows to significantly *reduce the computational complexity*, where for instance improvements are shown of 10% for a subset of three streams. The data rates are shown to follow a waterfilling type of allocation over time for the individual streams. Furthermore it has been shown that a *significant reduction in resource pool size* can be achieved compared to a baseline 'direct download' approach, where a potential reduction is demonstrated of up to 45%. Finally, it has been shown that transitioning from a traditional static approach (with a static fixed total system budget of 30%) to the proposed dynamic approach reduces the average required allocated cancellation taps with 68%.

References

[1] R. Cendrillon, G. Ginis, M. Moonen, and K. Van Acker, "Partial crosstalk precompensation in downstream VDSL," *Signal Processing*, vol. 84, no. 11, pp. 2005–201, Nov. 2004.

[2] 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*, vol. 2004, no. 10, pp. 1520–1535, Aug. 2004.

[3] P. Tsiaflakis, J. Vangorp, M. Moonen, J. Verlinden, and G. Ysebaert, "Partial crosstalk cancellation in a multi-user xDSL environment," in *IEEE International Conference on Communications*, Istanbul, Turkey, June 2006, vol. 7, pp. 3264–3269.

[4] A. R. Forouzan and L. M. Garth, "Computationally efficient partial crosstalk cancellation in fast time-varying DSL crosstalk environments," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 1–15, Jan. 2007.

[5] B. Li, P. Tsiaflakis, M. Moonen, J. Maes, and M. Guenach, "Dynamic partial crosstalk cancellation resource allocation algorithms for DSL access networks," *EURASIP Journal on Advances in Signal Processing*, vol. 150, pp. 1–29, Jul. 2012.

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

[7] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen, "Throughput and delay performance of DSL broadband access with cross-layer dynamic spectrum management," *IEEE Transactions on Communications*, vol. 60, no. 9, pp. 2700–2711, Sep. 2012.

[8] S.-H. Gary Chan and S.-H. Ivan Yeung, "Broadcasting video with the knowledge of user delay preference," *IEEE Transactions on Broadcasting*, vol. 49, no. 2, pp. 150–161, Jun. 2003.

[9] A. Hafid, "Providing a scalable video-on-demand system using future reservation of resources and multicast communications," in *5th International Workshop on Quality of Service Communications*, Columbia University, May 1997, pp. 1–12.

[10] J. Londono and A. Bestavros, "A two-tiered on-line server-side bandwidth reservation framework for the real-time delivery of multiple video streams," in *Proc. the SPIE and IS&T Conference on Multimedia Computing and Networking*, San Jose, CA, Jan. 2009, pp. 1–12.