Reprint

Dynamic Bit-loading in *pDSL* Communications Systems

N. Papandreou and Th. Antonakopoulos

The International Symposium on Power Lines Communications – ISPLC 2005

VANCOUVER, CANADA, APRIL 2005

Dynamic Bit-Loading in *pDSL* Communications Systems

Nikolaos Papandreou
Research Academic Computer Technology Institute
61 Riga Feraiou Str., 26100 Patras, Greece
E-mail: npapandr@cti.gr

Theodore Antonakopoulos
Dept. Electrical Engineering & Computers Technology
University of Patras, 26500 Rio - Patras, Greece
E-mail: antonako@ee.upatras.gr

***Abstract*— This paper presents a dynamic discrete bit-loading algorithm for multicarrier *pDSL* communication links that utilize the indoor power line network. In order to confront the time-varying behavior of power line transmission media, the proposed algorithm provides fast adaptive bit and power loading with reduced complexity using channel state information and a sub-channel link-allocation profile. Numerical results based on simulated network loading scenarios demonstrate the performance and the complexity of the proposed method.**

I. Introduction

The provision of broadband communications services over the indoor power grid has been studied using different approaches [1], [2]. The experience gained in wireless and digital subscriber line (DSL) communications has given rise to various multicarrier modulation (MCM) techniques for combating multipath fading and intersymbol interference in the power line communications (PLC) environment. The performance of MCM systems in a frequency-selective fading channel is limited by the time-varying behavior of its subchannels and thus, in order to confront this behavior, adaptive bit and power loading mechanisms have to be used. Several bit-loading algorithms have been proposed in the literature, e.g. [3], [4], which provide initial bit-allocation during the link establishment, while in [5] a dynamic bit-swapping scheme for time-varying MCM systems was described.

In [6], the concept of *pDSL* (power-line DSL) was presented as a new approach that converts the indoor power grid into a 'virtual binder' of high speed DSL-like MCM links. A *pDSL* network consists of multiple high-speed communication devices, called *pDSL* devices, and an internetworking device, called *pDSL* gateway, which acts as the central coordination unit of the indoor PLC network and also as its interface to the external communications infrastructure. An example of a *pDSL* network is illustrated in Figure 1.

In a *pDSL* network, a point-to-point link is established between each *pDSL* device and the *pDSL* gateway using non-overlapping frequency bands. Using the training procedures described in [6], an estimation of each link's response is calculated initially and is updated periodically based on measurements performed during the normal network operation. The links' responses are exploited by the *pDSL* gateway for generating the network's signal-to-noise ratio (SNR) profile. For improved system performance, dynamic bit-loading that is adapted to the network's time-varying behavior, has to be used at each active link.

This paper presents such a dynamic bit-loading method that is executed in the *pDSL* gateway and periodically updates the bit and power allocation profiles of each *pDSL* link. The proposed algorithm is based on a fast and optimum discrete bit-loading procedure [7] and uses the channel state information available at the *pDSL* gateway along with the results of a new bandwidth allocation method [8] that distributes the entire *pDSL* spectrum to all active links.

Section II discusses the bit-loading framework in the *pDSL* communications environment. Section III presents the dynamic discrete bit-loading algorithm, while Section IV provides simulation results that describe the performance of the dynamic bit-loading using an illustrative example and also provides numerical results that present the computational efficiency of the proposed method.

II. The Bit-Loading Framework of *pDSL*

A *pDSL* network that consists of a *pDSL* gateway and M *pDSL* devices forms M bidirectional point-to-point links, resulting to $2M$ 'virtual' links, i.e. M up-links and M down-links. The entire spectrum is divided into multiple, non-overlapping subchannels of fixed bandwidth and a subset of these subchannels is allocated to each link, with the requirement that each subchannel is assigned to one direction of transmission only. Therefore, the subchannel allocation and bit-loading processes have to satisfy two purposes: to distribute the subchannels to all active links and to calculate the bit and power distributions for the up-link and the down-link. We investigated the subchannel allocation problem in [8], where a Frequency Division Multiplexing (FDM) algorithm was proposed that provides fast subchannel allocation under a set of quantitative criteria.

Given the subchannel allocation of each link and the restriction for integer bit values, the optimum bit-loading in each direction follows the discrete greedy solution [4]. The dynamic bit-loading algorithm for fast adaptation of the bit and power allocations to the network SNR changes is presented in the next section and is based on a computationally efficient discrete loading process [7]. In particular, the method in [7] outperforms the greedy bit-loading schemes by exploiting an

0-7803-8844-5/05/$20.00

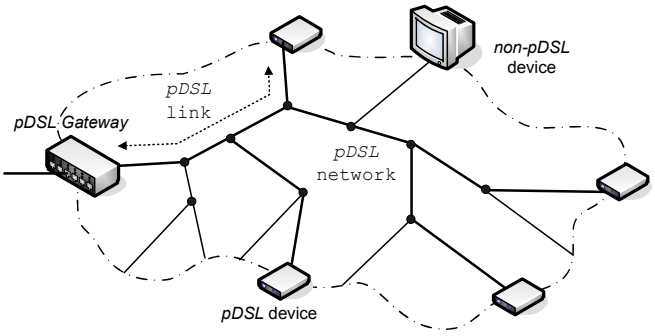

Fig. 1. The *pDSL* communications networking infrastructure.

initial bit profile and a multiple bits loading procedure that converges fast to the target-rate solution. In the following we consider the down-link transmission only and for simplicity we will use the term link for each 'virtual' down-link.

Before proceeding to the dynamic bit-loading algorithm, we introduce the network's subchannel allocation matrix that contains the subchannel allocation profile of each link and we formulate the discrete bit-loading problem.

A. The Network Subchannel Allocation Matrix

We denote as i the subchannel and j the *pDSL* link. Let $S = \{i : 1, 2, \ldots, N\}$ be the set of all subchannels in the *pDSL* network. Based on the results of the subchannel allocation algorithm [8] we denote as $S_j(t_k)$ the subset of subchannels in S allocated to link j at time t_k: $\bigcup_{j=1}^{M} S_j(t_k) \subseteq S$ and $\bigcap_{j=1}^{M} S_j(t_k) = \emptyset$, where M is the number of active links in the network at t_k.

A subchannel allocation matrix $\mathbf{F} = [f_{i,j}]_{M \times N}$ is generated at each network loading change indicating whether subchannel i belongs to link j according to the following rule, where $g_{i,j}$ denotes the subchannel gain-to-noise ratio:

1. `if` $i \notin S_j(t_k)$ `then`
2. $\quad f_{i,j} = 0$
3. `else if` $i \in S_j(t_k)$ `and` $i \notin S_j(t_{k-1})$ `then`
4. $\quad f_{i,j} = +1$
5. `else if` $i \in S_j(t_k)$ `and` $i \in S_j(t_{k-1})$ `then`
6. $\quad$ `if` $\frac{1}{\gamma_{i,j}(t_{k-1})} \leq \frac{g_{i,j}(t_k)}{g_{i,j}(t_{k-1})} \leq \frac{2^{b_{i,j}(t_{k-1})+1}-1}{2^{b_{i,j}(t_{k-1})}-1}$ `then`
7. $\quad\quad f_{i,j} = -1$
8. $\quad$ `else`
9. $\quad\quad f_{i,j} = +1$
10. $\quad$ `end if`
11. `end if`

The inequality condition in step 6 determines the bounds in order for the gain-to-noise ratio at time t_k to be considered invariant, i.e. the corresponding bit allocation $b_{i,j}$ is still supported. In particular, the lower bound depends on the subchannel margin $\gamma_{i,j}$ used for gain-to-noise ratio degradation immunity, while the upper bound describes the situation of gain-to-noise ratio improvement that enables one additional bit to be assigned.

B. The Loading Problem Formulation

Given the subchannel allocation $\mathbf{F}$, the discrete bit-loading problem for link j at time t_k is stated as follows:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i \in S_j(t_k)} P_{i,j} = P_j \\ \text{subject to} \quad & \textstyle\sum_{i \in S_j(t_k)} b_{i,j} = B_j^T, \\ & \textstyle\sum_{i \in S_j(t_k)} P_{i,j} \leq P_j^B, \\ & 0 \leq b_{i,j} \leq \overline{b}_{i,j}, \\ & b_{i,j} \in \mathbb{Z}_+ \end{aligned} \tag{1}$$

In (1), $b_{i,j}$ and $P_{i,j}$ are the number of bits and the power assigned to subchannel i respectively and are related by:

$$b_{i,j} = \log_2 \left(1 + \frac{P_{i,j} \cdot g_{i,j}}{\Gamma} \right) \tag{2}$$

where Γ is the SNR gap depending on the target error rate, the applied coding and the performance margin [9]. B_j^T and P_j^B are the required target-rate and the total available power budget of link j and $\overline{b}_{i,j}$ is the maximum number of bits that can be allocated in each subchannel according to:

$$\overline{b}_{i,j} = \min \left(b_{max}, \left\lfloor \log_2 \left(1 + \frac{\overline{P}_{i,j} \cdot g_{i,j}}{\Gamma} \right) \right\rfloor \right) \tag{3}$$

where b_{max} corresponds to the maximum allowable size of the signal constellations and $\overline{P}_{i,j}$ is the maximum allowable power imposed by the PSD guidelines of link j.

Every feasible bit-allocation is denoted as:

$$[b_{i,j}]_0^{\overline{b}_{i,j}} = \begin{cases} \overline{b}_{i,j}, & b_{i,j} \geq \overline{b}_{i,j} \\ b_{i,j}, & 0 < b_{i,j} < \overline{b}_{i,j} \\ 0, & b_{i,j} \leq 0 \end{cases} \tag{4}$$

From [7] the link's characteristic bit profile over the subchannels $i \in S_j(t_k)$ is defined as:

$$b'_{i,j} = \begin{cases} \lfloor \log_2(k_{i_{min},j}) + 1 \rfloor, & i = i_{max} \\ \lfloor \log_2(k_{i_{min},j}) \rfloor - \lfloor \log_2(k_{i,j}) \rfloor, & \text{otherwise} \end{cases} \tag{5}$$

where $k_{i,j} = \frac{g_{i_{max},j}}{g_{i,j}}$, $i_{max} = \arg_{i \in S_j(t_k)} \max(g_{i,j})$, $i_{min} = \arg_{i \in S_j(t_k)} \min(g_{i,j})$. The $b'_{i,j}$ profile is *efficient* according to [3] and therefore optimum.

The loading problem in (1) seeks for the power distribution of minimum total power that satisfies the target-rate. The difference between the power budget and the total power is exploited for increasing the system margin. The PLC network is a time-varying channel where strong SNR variation may be experienced and therefore large values of margin are required. In particular, the final margin in each subchannel is given in dB by:

$$\gamma_{i,j}(\text{dB}) = \gamma_m(\text{dB}) + 10 \log_{10} \left[\min \left(\frac{\overline{P}_{i,j}}{P_{i,j}}, \frac{P_j^B}{P_j} \right) \right] \tag{6}$$

where γ_m represents any minimum required margin inherited in the SNR gap.

The bit-loading problem in the *pDSL* network is related with two cases: the '*initial bit-loading*' case when a link is activated, and the '*dynamic bit-loading*' case that is executed periodically when new loading conditions exist. In the '*initial bit-loading*' case the solution to (1) is determined using [7]. As the network loading changes and a new subchannel allocation matrix $\mathbf{F}$ is generated, the '*dynamic bit-loading*' solution is calculated according to the algorithm of the next section.

The implementation of the algorithm requires the following matrices to be stored in the *pDSL* gateway:

- the network subchannel allocation matrix $\mathbf{F} = [f_{i,j}]_{M\times N}$
- the characteristic $k_{i,j}$ matrix $\mathbf{K} = [k_{i,j}]_{M\times N}$
- the characteristic bit profiles matrix $\mathbf{B'} = [b'_{i,j}]_{M\times N}$
- the maximum number of bits matrix $\overline{\mathbf{B}} = [\overline{b}_{i,j}]_{M\times N}$
- the network bit-allocation matrix $\mathbf{B} = [b_{i,j}]_{M\times N}$
- the network power-allocation matrix $\mathbf{P} = [P_{i,j}]_{M\times N}$

III. Dynamic Bit-Loading

The dynamic bit-loading is divided into three phases. In the first phase, the characteristic bit profile $b'_{i,j}$ of the link is updated according to the new $g_{i,j}$ values registered at the *pDSL* gateway. In the second phase, the bit profile is up/down shifted using multiple bits and converges to the target-rate solution with a maximum of one bit difference per subchannel. In the third phase, greedy bit-filling or bit-removal is performed for a limited number of bits.

Let $B_j = \sum_{i\in S_j(t_k)} [b_{i,j}]_0^{\overline{b}_{i,j}}$ and $\Delta_j = B_j^T - B_j$. Assuming that the network loading changed and the new subchannel allocation matrix $\mathbf{F}$ has been computed, the following three-phases procedure provides the new optimum bit-allocation for link j:

A. Initial Bit-Allocation

1. update $\overline{b}_{i,j}$, $\forall i : f_{i,j} = +1$
2. update $k_{i,j}$ and $b'_{i,j}$, $\forall i : f_{i,j} = +1$
3. if $b'_{i_{max},j} > \overline{b}_{i_{max},j}$ then
4. $\quad b'_{i,j} = b'_{i,j} - (b'_{i_{max},j} - \overline{b}_{i_{max},j})$, $\forall i : f_{i,j} \neq 0$
5. end if
6. initialize $b_{i,j} = b'_{i,j}$, $\forall i : f_{i,j} \neq 0$
7. calculate B_j and Δ_j
8. if $\Delta_j > 0$ then
9. $\quad$ FLAG = ADD
10. else
11. $\quad$ FLAG = REM
12. end if

B. Multiple Bits Allocation

1. while TRUE do
2. $\quad$ if FLAG = ADD then
3. $\quad\quad \mathcal{I} = \{i : f_{i,j} \neq 0, \overline{b}_{i,j} > b_{i,j} \geq 0\}$ and $a = \lfloor \frac{|\Delta_j|}{\mathcal{P}\{\mathcal{I}\}} \rfloor$
4. $\quad$ else
5. $\quad\quad \mathcal{I} = \{i : f_{i,j} \neq 0,\ b_{i,j} > 0\}$ and $a = -\lfloor \frac{|\Delta_j|}{\mathcal{P}\{\mathcal{I}\}} \rfloor$
6. $\quad$ end if
7. $\quad$ if $a = 0$ then
8. $\quad\quad$ goto FINAL BIT-ALLOCATION
9. $\quad$ end if
10. $\quad$ set $b_{i,j} = b_{i,j} + a$, $\forall i : f_{i,j} \neq 0$
11. $\quad$ calculate B_j and Δ_j
12. end while

C. Final Bit-Allocation

1. if FLAG = ADD then
2. $\quad$ do GREEDY BIT-FILLING in $\mathcal{I}$ for $|\Delta_j|$ bits
3. else
4. $\quad$ do GREEDY BIT-REMOVAL in $\mathcal{I}$ for $|\Delta_j|$ bits
5. end if

At the end of the process the new bit-loading for link j is:

$$b_{i,j} = \begin{cases} [b_{i,j}]_0^{\overline{b}_{i,j}}, & \text{if } f_{i,j} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

With reference to the above algorithm some remarks are needed for clarification:

- Step A.3 is used in order to move the characteristic bit profile (5) within the valid bit limits [7].
- In steps B.3 and B.5, we denote as $\mathcal{P}\{\mathcal{I}\}$ the population of the subset $\mathcal{I}$, while the *floor* $\lfloor\cdot\rfloor$ operations guarantee that a converges to 0.
- If the bit-allocation $b_{i,j}$ in step B.3 contains negative values, as a result of previous down-shifts, then $a = \min\{\lfloor \frac{|\Delta_j|}{\mathcal{P}\{\mathcal{I}\}} \rfloor, a^*\}$, where $a^* = \min\{|b_{i,j}|\}$, $\forall i : f_{i,j} \neq 0$ and $b_{i,j} < 0$, see [7] for more details.
- If Δ becomes zero in steps A.7 or B.11, then the algorithm is completed with the bit-allocation profile calculated so far.
- If $f_{i_{min},j} = 1$ or $f_{i_{max},j} = 1$, then step A.2 is performed $\forall i : f_{i,j} \neq 0$. In this case the dynamic bit-loading algorithm has the maximum computational load.
- Given the new bit-allocation (7) for link j, the corresponding power-allocation profile is calculated using (2) and the margin in (6).

IV. Simulation Results

For the simulation results presented in this section, we consider the network example of [6] and we assume that the *pDSL* gateway is connected to the termination point T_1, while there are three *pDSL* devices which are corrected to T_2, T_4 and T_6. We also consider that there are 64 available subchannels and we demonstrate results of two loading scenarios.

Scenario ♯1 is used to describe the loading conditions when the network is activated, while scenario ♯2 represents the loading conditions the first time the dynamic bit-loading algorithm is executed. All *pDSL* equipment exhibit impedances of 100 Ohms and the remaining termination impedances vary according to the loading scenario. In scenario ♯1 all the remaining terminations are open circuits, while in scenario ♯2 a $10\angle -90^o$ Ohms impedance is connected at T_5.

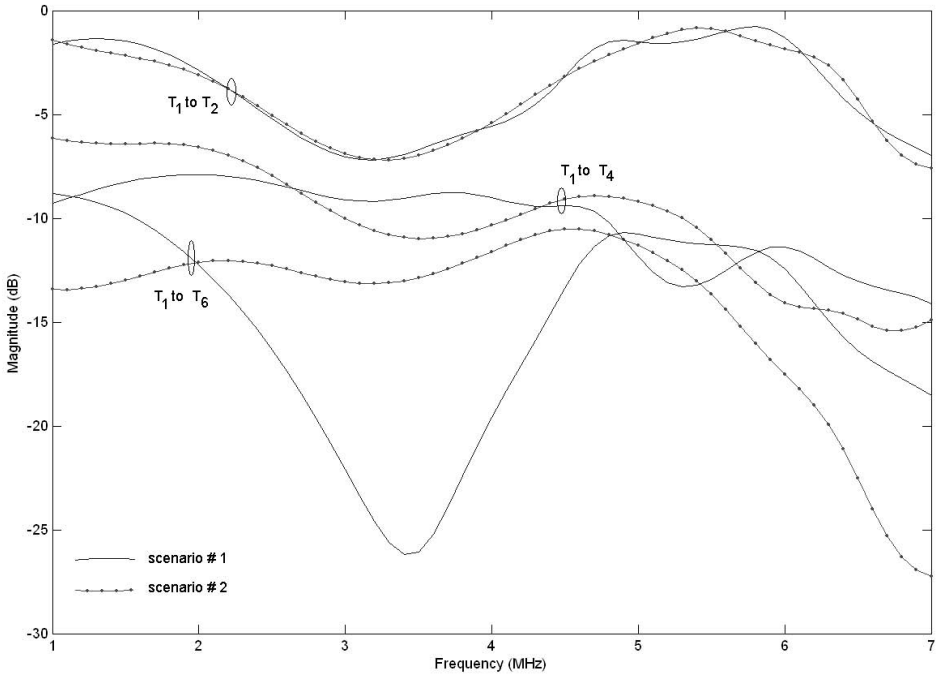

Fig. 2. The transfer function of the three links under different loading conditions.

Figure 2 shows the magnitude of the transfer function of all links in both scenarios. We observe that the magnitude of all links changes, even when a single termination impedance changes in the network.

Figures 3, 4 and 5 present the bit and subchannel allocations for each link and for each network loading scenario. The subchannel allocation results are based on the algorithm in [8]. In Figure 3 we see that link $T_1 - T_2$ is assigned the same subchannels in both cases due to the insignificant transfer function change. This result can be verified by the transfer function plots in Figure 2. On the other hand, in Figures 4 and 5 we see that the links $T_1 - T_4$ and $T_1 - T_6$ have different subchannel allocation profiles for the two scenarios, although some subchannels are assigned to the same link in both cases, i.e. at the network activation and at the network loading change.

In Table I we present some numerical results that demonstrate the complexity of the dynamic bit-loading algorithm proposed in the previous section. The proposed algorithm is compared with the greedy bit-swapping method of [5]. For each link, Table I presents the target rate in bits per symbol, as well as the total number of bits that have to be swapped by the algorithm in [5] as soon as the network loading changes, including any final bit-filling operations so as to achieve the target-rate (see [5] for details). In order to measure the performance of the proposed method, both algorithms were implemented in a 900 Mflops DSP.

TABLE I
PERFORMANCE OF THE DYNAMIC BIT-LOADING ALGORITHM

Link	T_1 to T_2	T_1 to T_4	T_1 to T_6
Target-rate (bpsym)	220	220	230
Total bits swapped	0	137	154
	Execution time (msec)		
proposed algorithm	1.43	1.45	2.11
algorithm of [5]	1.03	4.32	5.51
Improvement factor	0.72	2.98	2.61

Table I presents the total execution time of each algorithm (in msec), as well as the improvement factor (the ratio of the corresponding execution times) regarding the computational complexity of the proposed algorithm compared with the algorithm in [5].

We remark that, although no bits have to be swapped for link $T_1 - T_2$, the algorithm in [5] has a minimum execution time due to the inherited control functions. As the number of bits that have to be swapped increases, the computational load of [5] also increases. On the other hand, due to the fact that the bits remained to be allocated in the third phase of the dynamic algorithm changes with modulo $\mathcal{P}\{\mathcal{I}\}$, the computational load of the proposed method is limited. The improvement factor for link $T_1 - T_2$ is less than unity, while

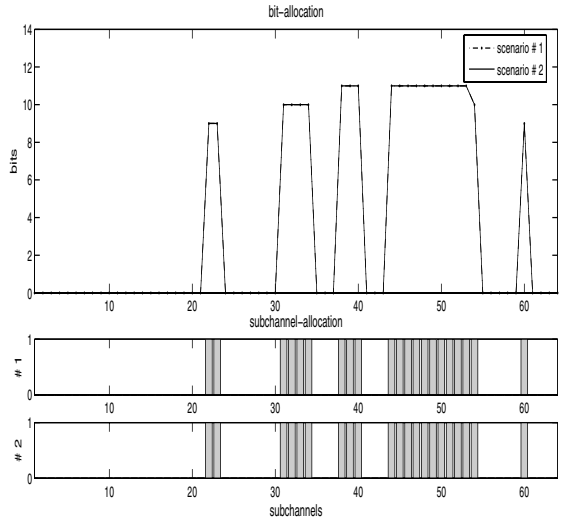

Fig. 3. The bit and subchannel allocations of link $T_1 - T_2$.

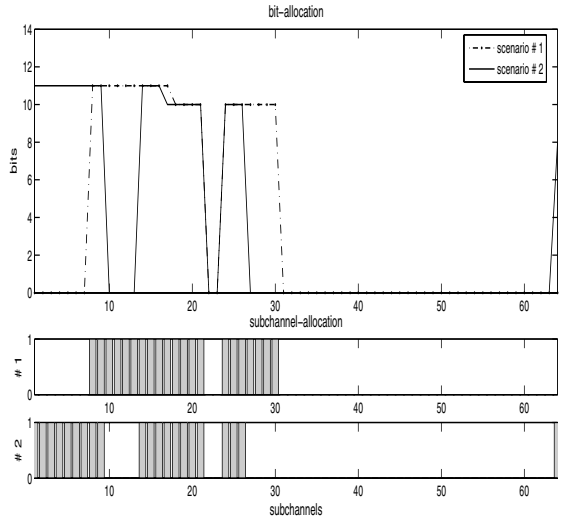

Fig. 4. The bit and subchannel allocations of link $T_1 - T_4$.

for the other links the dynamic algorithm is executed almost 3 times faster than the greedy bit-swapping. Note, that the implementation of the dynamic algorithm did not encounter any margin immunity and as a result step A.2 in the previous section was executed $\forall i : f_{i,j} \neq 0$ for all links, thus increasing the total computational load.

In general, as the number of bits that need to be swapped in order to update the bit-allocation of a link increases, the computational load of the bit-swapping process increases substantially. In a slow time-varying channel the greedy bit-swapping method may provide fast results, however in the PLC environment the channel may exhibit significant gain-to-noise ratio change, that requires a large number of bits to be reallocated. Moreover, given a large number of subchannels,

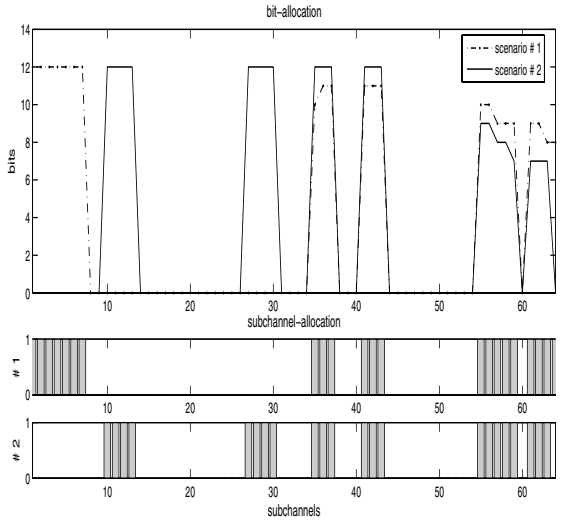

Fig. 5. The bit and subchannel allocations of link $T_1 - T_6$.

the greedy bit-swapping operations will increase the final complexity. On the other hand, the proposed dynamic method is based on a fast and optimum discrete bit-loading process and results to reduced complexity.

V. Conclusions

In this work we presented a dynamic discrete bit-loading algorithm for the *pDSL* networking environment, that is executed at the *pDSL* gateway and periodically updates the bit and power allocations of all active links. The new algorithm uses a fast and optimum bit-loading process and is based on channel state information and on a subchannel allocation profile for each link that is calculated periodically at the *pDSL* gateway. The proposed bit-loading algorithm provides low complexity compared with the greedy bit-swapping method.

References

[1] W. Liu, H. Widmer, and P. Raffin, "Broadband PLC access systems and field deployment in European power line networks," *IEEE Commun. Mag.*, vol. 41, no. 5, pp. 114–118, May 2003.

[2] E. Biglieri, "Coding and modulation for a horrible channel," *IEEE Commun. Mag.*, vol. 41, no. 5, pp. 92–98, May 2003.

[3] J. Campello, "Practical bit loading for DMT," in *Proc. IEEE ICC'99*, June 1999, vol. 2, pp. 801–805.

[4] A. Fasano, G. Di Blasio, E. Baccarelli, and M. Biagi, "Optimal discrete bit loading for DMT based constrained multicarrier systems," in *Proc. IEEE ISIT'02*, July 2002, p. 243.

[5] A. Fasano and G. Di Blasio, "Optimal discrete dynamic loading algorithms for multicarrier systems," in *Proc. IEEE SPAWC'03*, June 2003, pp. 373–377.

[6] D. Anastasiadou and T. Antonakopoulos, "Broadband communications in the indoor power line environment: The *pDSL* concept," in *Proc. ISPLC'04*, Zaragoza, Spain, March 2004, pp. 334–339.

[7] N. Papandreou and T. Antonakopoulos, "A new computationally efficient discrete bit-loading algorithm for DMT applications," *accepted for publication in the IEEE Trans. Commun.*

[8] T. Antonakopoulos and N. Papandreou, "Subchannel allocation on multiple *pDSL* lines," in *Proc. ISPLC'05*, Vancouver, Canada, April 2005.

[9] J. M. Cioffi, "A multicarrier primer," ANSI Contribution T1E1.4/91-157, Clearfield, Florida, Nov. 1991.