On the Analytical Performance Optimization of Wireless Data Broadcasting

C. K. Liaskos, *Member, IEEE,* S. G. Petridou, *Member, IEEE,* G. I. Papadimitriou, *Senior Member, IEEE,* P. Nicopolitidis, *Member, IEEE,* and A. S. Pomportsis

***Abstract*—The role of the push-based architecture in modern wireless telecommunication systems has been gaining momentum over the past years. Yet these systems pose a challenging problem: the creation of the optimal common broadcast schedule that best fits the needs of the totality of the available clients. To this end various approaches have been proposed, one of the most well-known and widely accepted being the Broadcast Disks method. This method ensures the periodicity and proportionality attributes of the broadcast schedule, but leaves ample room for performance optimization, a task traditionally assigned to heuristic algorithms. This paper presents a mathematical analysis of the Broadcast Disk method, which establishes a new analytical performance optimization procedure. Comparison with other related well-known algorithms yielded better performance in every client test case. Finally, the analysis showed that even small deviations from optimal parameter values may lead to significant performance degradation in terms of clients' mean waiting time.**

***Index Terms*—Analytical approach, broadcast disks, mean waiting time, wireless push system.**

I. Introduction

RECENT years have witnessed the wide-spreading use of wireless push-based systems. Simple in architecture and implementation, lightweight and energy efficient - especially from the client's point of view and both hardware and software wise - the push-based approach has been adopted for use in a variety of information dissemination applications and has been incorporated in almost every single mobile telecommunication device. Popular uses include airport and hospital informative systems, instant messaging services, and multimedia on demand over the internet or cellular networks. Consequently, the on-growing interest of the telecommunications industry has spurred the research on the performance optimization of these systems.

The objective of the aforementioned research is easy to define, yet difficult to achieve. The goal is to produce a common broadcast schedule that minimizes the mean waiting time of the totality of the wireless clients. Ideally, a straightforward mathematical analysis of the system would evince the aforementioned procedure, yet it has been proven that this approach falls into the NP-hard category of problems [1–4], rendering it impossible to solve with the current mathematical tools.

Manuscript received December 19, 2008.

C.K. Liaskos, S.G. Petridou, G.I. Papadimitriou, P. Nicopolitidis and A.S. Pomportsis are with the Department of Informatics, Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece (e-mails: {cliaskos, spetrido, gp, petros, apombo}@csd.auth.gr)

A. Related Work

In order to overcome this dead-end, less constricting approaches were adopted, one of the most popular and influential being the Broadcast Disks method [5]. This method can be briefly described as a generic broadcast scheduling framework that ensures the periodicity and proportionality of the transmission schedule of a push-based system: the interval between two consecutive occurrences of the same data item in the schedule is held constant and the total number of its appearances proportional to its popularity. These broadcast schedule attributes have been proven to be very practical and thus the Broadcast Disks method has constituted the basis for many subsequent studies involving the pre-fetching, caching [6, 7] and indexing of data [8, 9], hybrid data delivery [10, 11] and QoS-related broadcast scheduling strategies [12–15].

Performance enhancement-related research aiming to minimize the clients' waiting time has until recently been epitomized by the GREEDY algorithm [16], which superseded the approaches presented in [17, 18]. These algorithms relied on empirical observations. On the same notion, Clustering-driven Wireless Data Broadcasting (CWDB) procedure [19] surpassed the GREEDY algorithm in the majority of the test cases. Furthermore, it proved that any performance comparison between broadcast scheduling algorithms must implicate a wide range of client cases in order to be credible.

In [20] analytical tools for pinpointing the optimal system parameters were presented, a task previously requiring time consuming simulations [16, 19]. Comparison with the GREEDY algorithm yielded promising results, with the analysis-derived procedure outperforming the GREEDY algorithm in the 87% out of a total 54 client cases that were tested. This initial analysis assumed uniform workload distributions, a simplification that hindered it from performing better in the remaining client cases. However, the analysis' potential was promising, and thus motivated further research.

B. Contribution

This paper presents a mathematical analysis of the Broadcast Disks method, overcoming the simplifications of [20] i.e. the non-parametrical workload distribution functions. As a practical application of this analysis, a new broadcast scheduling procedure is presented, namely the Optimization-based Broadcast Scheduling Procedure (OBSP).

The new procedure is compared with other well-known alternatives, namely the GREEDY-based Broadcast Scheduling Procedure (GBSP) and the Clustering-driven Wireless Data Broadcasting (CWDB) algorithm. To the best of our knowledge, these are, performance-wise, the most efficient of the family of algorithms that refer to the Broadcast Disks method. Furthermore, analytically-derived client waiting times are compared with their corresponding simulation-derived values as a means of ensuring the validity of the theoretical analysis.

Finally, it is shown that even small deviations of the system's parameters from their optimal values may lead to significant performance degradation in terms of clients' mean waiting time.

It must be clarified than in the context of this paper the terms "server's mean response time" and "client's mean waiting time" are equivalent and refer to the mean duration of the interval between the moment a need for a specific data item arises in the client, and the moment this item is broadcasted by the server. This is to compensate for the fact that the concepts of "client's query" and "server's answer" are inexistent in push-based systems.

C. Roadmap

The remainder of this paper is organized as follows: Section II familiarizes the reader with the Network Architecture and terminology of push-based systems. In Section III the mathematical analysis of the Broadcast Disks method is presented, and the OBSP is formulated. Section IV deals with the comparison between the OBSP, GBSP and CWDB procedure. Simulation configurations, results and remarks are given in the appropriate subsections. Finally, conclusion is given in Section V.

II. Theoretical Background

A. Network Architecture and Operation

The physical part of the network under discussion-being a typical push-based system-consists of a server, a database, one or more clients and an asymmetric communication channel. The database [21] containing $DBSize$ data items - most commonly known as pages - is connected to the server. All pages are considered to be of equal size. The server is properly equipped (hardware and protocol wise) [14] in order to act as a central node of a cellular network. Any adequately equipped portable device can then act as a client.

The server is either assumed to know the clients' preferences in advance in case they are static, or capable of acquiring this information dynamically by means of a lightweight feedback mechanism [15, 19]. Once sufficient information of the clients' preferences has been acquired, the server proceeds to create a fitting broadcast schedule for the $DBSize$ pages. This schedule is then forwarded to the transmitting device, and eventually to the clients.

The communication channel is asymmetric, either because of the electromagnetic properties of the transmission medium or as a consequence of hardware limitations. From a client's point of view this stands for adequate download bandwidth, and a very limited - but not zero - upload capability. The assumption that the clients are capable of transmitting data must not be considered as an aberration from the push prototype. It is rather a necessary condition for the existence and operation of any feedback operation, as described above.

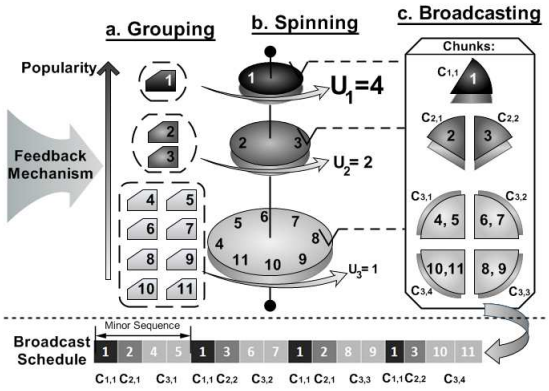


Fig. 1. Broadcast Disks method overview

Research on feedback systems and learning automata [15, 19] constitutes a closely related yet separate field. In the context of this paper, it is assumed that the server has already acquired a sufficient degree of knowledge of the clients' preferences.

B. The Broadcast Disks Method

The Broadcast Disks method is a generally approved framework for data broadcast scheduling in push-based wireless network configurations [22]. The method assumes that the popularity of each data item is known, most commonly in the form of their request probability. This task is assigned to the system's feedback mechanism. The Broadcast Disks method then intervenes to create a broadcast schedule which complies with two simple rules. Firstly, the number of occurrences of each page in a single broadcast schedule must be proportional to its request probability. Secondly, all instances of the same page must be uniformly distributed inside the broadcast schedule in order to avoid the famous "bus stop" paradox [23]. This is accomplished in three steps as depicted in Fig. 1:

1) *Grouping*: the pages are grouped according to their popularity in a fixed number of groups called disks. This scheme can be abstractedly represented as an array of physical disks, spinning around a common axis.
2) *Spinning*: each of these disks is then set to spin with an angular velocity proportional to the aggregate demand of its contained pages.
3) *Broadcasting*: finally, an imaginary set of stationary heads retrieves pages from the disks and forwards them to the broadcasting system in the same order that they have been read.

The method does not state how the tasks of *Grouping* and *Spinning* are to be accomplished. Thus, the optimization's goal is to define:

1) the optimal number of disks, NoD

2) the number of pages that each disk contains, $DiskSize(i),\ i = 1, \ldots, NoD$
3) the optimal spinning velocity of each disk, $U_i,\ i = 1, \ldots, NoD$.

However, the method defines how the serialization task of *Broadcasting* is to be accomplished, in great detail [5]. More specifically, it requires the serialization task to take place according to the Algorithm 1, where $C_{i,j}$ refers to the j^{th} chunk of the i^{th} disk. An example of the process described by Algorithm 1 is illustrated in Fig. 1.

Algorithm 1 The default broadcast schedule constructor algorithm of the Broadcast Disks method.

1: Calculate the max_chunks as the Least Common Multiple of the disks' velocities U_i, where $i = 1, \ldots, NoD$
2: Partition each disk into $num\_chunks(i) = max\_chunks/U_i$ chunks
3: Broadcast the $C_{i,j}$ chunks in a round robin manner

Finally, it is important to calculate the length L of the broadcast schedule. It holds that $L = max\_chunks \cdot M$, where M denotes the length of the minor sequence, as presented in Fig. 1. Furthermore, it stands that $M = \sum_1^{NoD} chunk\_size(i)$, where $chunk\_size(i)$ denotes the size of each chunk of the i^{th} disk. Thus, since $chunk\_size(i) = \lceil \frac{DiskSize(i)}{num\_chunks(i)} \rceil$ we conclude that:

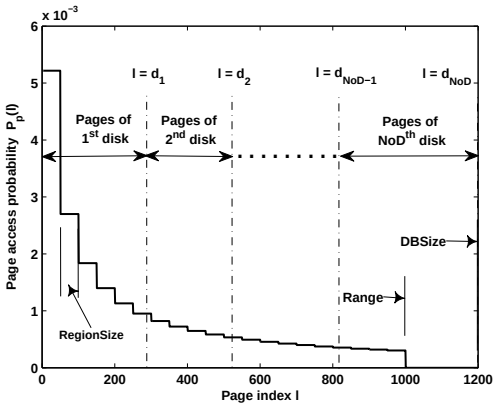
$$L = max\_chunks \cdot \sum_{i=1}^{NoD} \lceil \frac{DiskSize(i)}{num\_chunks(i)} \rceil \qquad (1)$$

C. Equivalent Client Probabilistic Model

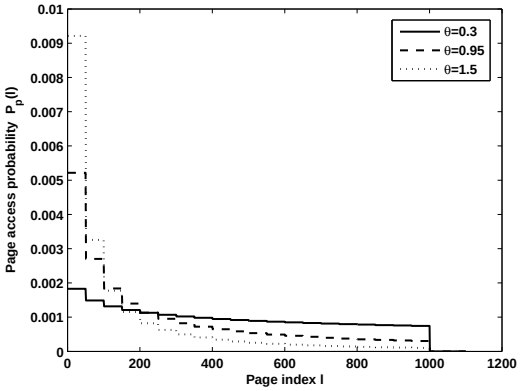
Concerning the client probabilistic model, the present work is in accordance with the assumptions made in preceding papers [5, 16]. According to these studies, the push server has no means of differentiating between the clients, as this would degrade the system's architecture into a classic client-server (pull) scheme. Therefore, for the analysis' purposes, it is possible to adopt an equivalent, single-client model [5, 16]. This equivalent client presents the same probabilistic behavior as the clients that he substitutes.

The equivalent client's p.d.f. P_p has typically the form of Fig. 2(a). More specifically, the client is considered to access only a random subset of the server's $DBSize$ pages, and their number will be denoted as $Range$. These pages are then organized further in equally sized r groups called Regions. All $RegionSize$ pages in one Region have equal probability of being requested by the client. The Regions themselves are considered to follow the zipfian probability distribution [24], according to which $P(k) \propto \frac{1}{k^\theta}$, where $k = 1, \ldots, r$ and θ is the zipfian p.d.f. skewness parameter (and as such $\theta \neq 1$).

Physically the θ parameter can serve primarily as an indication of the degree of similarity of the clients' preferences. As shown in Fig. 2(b), higher θ values indicate the existence of an increasingly popular team of pages. Therefore, the clients' preferences converge. From another point of view, θ can express the degree of correlation of the data items. Highly correlated data items would result into large teams of pages being accessed per single, initial client query. Thus, highly selective schemes, as the one depicted in Fig 2(b) for $\theta = 1.5$, would be less probable to occur. Hence, higher θ values may indicate less correlated data items.



(a) The client's page probability distribution function



(b) Effect of the θ parameter on the client's p.d.f

Fig. 2. Client probabilistic model

Finally, with the total number of pages in the server's database $DBSize$ being static, the client's page p.d.f. illustrated in Fig. 2(a) is fully defined by the three parameters $\{Range, RegionSize, \theta\}$. Thus, each different combination of these parameters represents a distinct client case.

III. Mathematical Analysis

Assume that the cyclically repeated broadcast program has been constructed and that it comprises of L equally sized transmissions of pages. If a page with index $l \in [1, DBSize]$ originated from a disk with an angular velocity of $U_i,\ (i = 1, \ldots, NoD)$, it will appear exactly U_i times inside the schedule, provided that the velocity of the last and slowest disk is equal to one unit. Uniform distribution of all same page appearances inside the schedule is ensured through the use of the Broadcast Disks method, as stated already. Thus the mean waiting time of a client expecting a page l belonging to the i^{th} disk with velocity U_i will, according to [23], be:

$$D(l) = \frac{L}{2U_i} \qquad (2)$$

Additionally, it is assumed that enough time has elapsed for the system to acquire an adequate approximation of the $Range$, $RegionSize$ and θ (zipf p.d.f.) parameters through the feedback mechanism described earlier. Thus the server has sufficient knowledge of the client's page p.d.f, whose general form is depicted in Fig. 2(a) and is denoted as $P_P(l)$.

The upcoming grouping of pages into a number of NoD disks is tantamount to choosing the segmentation points denoted as $d_i,\ i = 1, \ldots, NoD$ in Fig. 2(a). After the server pages have been sorted in descending access probability, these points signify the last page of each disk. In order to define the d_i points, the following approaches will be examined:

1) Perform the grouping at a level of whole regions, assuming that equiprobable pages should not be split over different disks. In other words the d_i segmentation points can only be placed at the end of a $Region$.
2) Form the last and slowest moving disk deterministically by grouping together the pages with null popularity. This approach derives from the logical assumption that the useless pages should be separated from the useful ones, and not be broadcasted more than once per broadcast period. Thus, $d_{NoD-1} = Range$ and $d_{NoD} = DBSize$.

By following or not the above approaches, four analysis variations can take place:

1) Adopt none of them. This case will be denoted as Non-Deterministic, Page-Grouping or NDPG-Variant.
2) Adopt only the first of them (Non-Deterministic, Region-Grouping or NDRG-Variant case).
3) Adopt only the second of them (Deterministic, Page-Grouping or DPG-Variant case).
4) Adopt both of them (Deterministic, Region-Grouping or DRG-Variant case).

A. Non-Deterministic, Page-Grouping (NDPG-Variant)

Consider a series of N queries of the equivalent client. Out of these N queries, exactly:

$$N_1 = N \cdot \sum_{l=1}^{d_1} P_P(l) \tag{3}$$

refer to pages that belong to the first disk. By setting:

$$G(x) = \sum_{l=1}^{x} P_P(l) \tag{4}$$

Equation 3 can be rewritten as:

$$N_1 = N \cdot G(d_1) \tag{5}$$

For the remaining disks we have:

$$N_i = N \cdot [G(d_i) - G(d_{i-1})], \quad i = 2, \ldots, (NoD - 1) \tag{6}$$

and

$$N_{NoD} = N \cdot [G(d_{NoD}) - G(d_{NoD-1})] = \\ = N \cdot [G(d_{DBSize}) - G(d_{NoD-1})] = N \cdot [1 - G(d_{NoD-1})] \tag{7}$$

From (2) is derived that the aggregate waiting time W_i for pages belonging to the i^{th} disk is:

$$W_i = N_i \cdot \frac{L}{2U_i} \tag{8}$$

W_i will be referred to as the i^{th} disk's workload. The client's mean waiting time can then be calculated as:

$$\overline{D} = \frac{\sum_{i=1}^{NoD} W_i}{N} = \frac{\sum_{i=1}^{NoD} N_i \frac{L}{2U_i}}{N} = \\ = \frac{L}{2} \cdot \left[\frac{G(d_1)}{U_1} + \frac{G(d_2) - G(d_1)}{U_2} + \cdots + \frac{1 - G(d_{NoD-1})}{U_{NoD}} \right] \tag{10}$$

A major assumption concerning the disks' workloads is that they are not uncorrelated, but instead follow specific distributions. Thus, since in the most general case stands that

$$p_1 \cdot W_1 = p_2 \cdot W_2 = \cdots = p_{NoD} \cdot W_{NoD},\ p_i \in R,\ i = 1 \ldots NoD \tag{11}$$

we assume that the workloads can follow either:

1) the progressively ascending fashion depicted in Fig. 3(a), and thus:

$$p_i^{ASC} = (1 + NoD - i)^{\varphi},\ i = 1, \ldots, NoD,\ \varphi \in [0, \infty) \tag{12}$$

2) the progressively descending fashion depicted in Fig. 3(b), and in this case:

$$p_i^{DESC} = i^{\varphi},\ i = 1, \ldots, NoD,\ \varphi \in [0, \infty) \tag{13}$$

3) the hill-like scheme depicted in Fig. 3(c), where:

$$p_i^{HILL} = (1 + \lceil \tfrac{NoD}{2} \rceil - i)^{\varphi}, for\ i = 1 \ldots \lceil \tfrac{NoD}{2} \rceil\ or \\ p_i^{HILL} = (1 - \lceil \tfrac{NoD}{2} \rceil + i)^{\varphi}, for\ i = \lceil \tfrac{NoD}{2} \rceil + 1 \ldots NoD \tag{14}$$

when NoD is odd and $\varphi \in [0, \infty)$, and:

$$p_i^{HILL} = (1 + \tfrac{NoD}{2} - i)^{\varphi}, for\ i = 1 \ldots \tfrac{NoD}{2}\ or \\ p_i^{HILL} = (1 - \tfrac{NoD}{2})^{\varphi}, for\ i = \tfrac{NoD}{2} + 1 \ldots NoD \tag{15}$$

when NoD is even and $\varphi \in [0, \infty)$.

In all three cases, a parameter φ is introduced in order to control the steepness of the disks' workload distribution. For $\varphi = 0$ all disks have equal workloads (uniform distribution) and the distributions become progressively more steep as φ increases. It is also clear that p_i depends only on φ and NoD:

$$p_i = f(\varphi, NoD) \tag{16}$$

In any case, (3) can be rewritten through (8) as:

$$p_1 \frac{L \cdot N}{2} \cdot \frac{G(d_1)}{U_1} = p_2 \frac{L \cdot N}{2} \cdot \frac{G(d_2) - G(d_1)}{U_2} = \ldots \\ \ldots = p_{NoD} \frac{L \cdot N}{2} \cdot \frac{1 - G(d_{NoD-1})}{U_{NoD}} \tag{17}$$

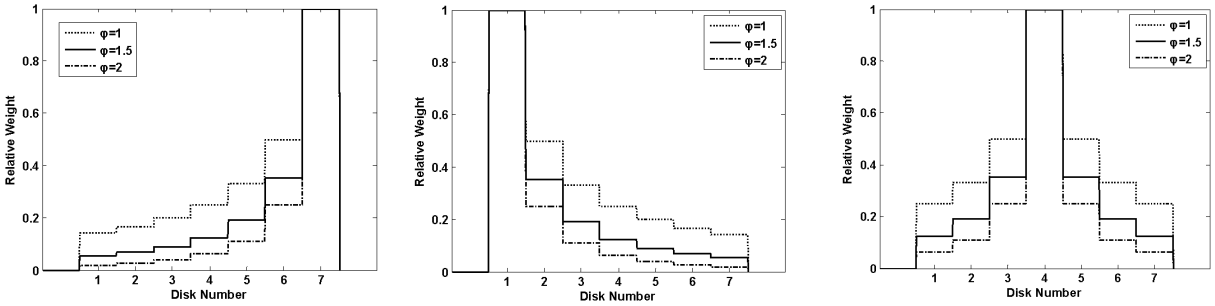
which can be simplified as:

$$p_1 \frac{G(d_1)}{U_1} = p_2 \frac{G(d_2) - G(d_1)}{U_2} = \ldots = p_{NoD} \frac{1 - G(d_{NoD-1})}{U_{NoD}} \tag{18}$$

Considering the U_i and p_i as simple parameters and the $G(d_i)$ as unknown variables, (18) represents a linear $(NoD - 1) \times (NoD - 1)$ system which can be easily expressed in matrix form $(A \cdot X = B)$ as presented in (9).

The parameter matrix A of the linear system expressed by (9) is a symmetric tridiagonal one. Furthermore it satisfies the criteria:

$$|a_{1,1}| \geq |a_{1,2}| \\ |a_{i,i}| \geq |a_{i,i-1}| + |a_{i,i+1}| \quad i = 2 \ldots n - 1 \\ |a_{n,n}| \geq |a_{n,n-1}| \tag{19}$$



(a) Ascending workload distribution (b) Descending workload distribution (c) "Hill" workload distribution

Fig. 3. Disks' workload distributions

$$\begin{pmatrix} \frac{p_1}{U_1}+\frac{p_2}{U_2} & -\frac{p_2}{U_2} & 0 & 0 & \dots & 0 & 0 & 0 \\ -\frac{p_2}{U_2} & \frac{p_2}{U_2}+\frac{p_3}{U_3} & -\frac{p_3}{U_3} & 0 & \dots & 0 & 0 & 0 \\ 0 & -\frac{p_3}{U_3} & \frac{p_3}{U_3}+\frac{p_4}{U_4} & -\frac{p_4}{U_4} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & -\frac{p_{NoD-1}}{U_{NoD-1}} & \frac{p_{NoD-1}}{U_{NoD-1}}+\frac{p_{NoD}}{U_{NoD}} \end{pmatrix} \cdot \begin{pmatrix} G(d_1) \\ G(d_2) \\ G(d_3) \\ \vdots \\ G(d_{NoD-1}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \frac{p_{NoD}}{UNoD} \end{pmatrix} \quad (9)$$

Thus the system always has a unique solution [25]. In addition, the convergence to this solution is known to be very quick, requiring $O(NoD)$ operations [25], with the optimal NoD parameter itself typically ranging from 1 to 10.

The velocities themselves are usually set as arithmetic progression terms [16], and as such:

$$U_i = (NoD - i) \cdot \Delta + 1,\ i = 1 \dots NoD \quad (20)$$

where Δ denotes the series' common difference. An alternative way to accomplish the same task, which was introduced in [19], is to set $U_{NoD} = 1$ deterministically, and then calculate the remaining terms as follows:

$$U_i = U \cdot [(NoD - 1 - i) \cdot \Delta + 1],\ i = 1 \dots NoD - 1 \quad (21)$$

where U an arbitrarily chosen parameter.

Finally it must be stated that since the velocities U_i all derive from either (20) or (21). Furthermore, concerning the workload distribution parameters p_i, (16) is in commission. Thus, the only parameters actually requiring definition are the common difference Δ (or the pair $\{\Delta, U\}$ in case of (21)) and the steepness regulator φ.

Once the system (9) is solved, the client's mean waiting time can be estimated through (10). Furthermore, from the resulting $G(d_i)$ values the corresponding d_i can be easily obtained, since $G(x)$ is a strictly increasing function and thus one-to-one and reversible. Calculating the corresponding disk sizes is then a trivial task.

B. Non-Deterministic, Range-Grouping (NDRG-Variant)

Performing the grouping at a level of whole regions instead of pages is - as stated earlier - equivalent to placing the d_i segmentation points at the end of $Regions$. No further analysis is required, except for replacing the d_i points obtained from the NDPG variant with their nearest Region end-points in the following manner:

$$d_i \rightarrow round(\frac{d_i}{RegionSize}) \cdot RegionSize,\ i = 1 \dots NoD - 1 \quad (22)$$

Care must be taken of the new d_i points in order to meet the criteria: $d_1 \neq 1$, $d_i \neq d_j\ \forall i, j \in [1, NoD]\ and\ i \neq j$, $d_i < d_{NoD}\ \forall i$

Inability to comply with these criteria yields failure of the method and a decrease of the NoD parameter by at least one unit is then imperative.

C. Deterministic, Page-Grouping (DPG-Variant)

In this case, the zero voted pages deterministically form the last and slowest moving disk. Thus, it stands that $d_{NoD-1} = Range$ and $d_{NoD} = DBSize$. The $(NoD-1) \times (NoD-1)$ linear system (9) is then reduced to the $(NoD-2) \times (NoD-2)$ system of (23). The procedure then continues as described in Section III-A.

D. Deterministic, Region-Grouping (DRG-Variant)

This analysis variation simply uses the d_i values produced by system (23) and reassigns them to their nearest $Region$ end-points, exactly as NDRG does with the results of NDPG. The same criteria as NDRG must be met as well.

E. Optimization-based Broadcast Scheduling Procedure (OBSP)

At this point, the client's mean waiting time optimization procedure can be fully defined as a series of simple steps described below:

1) A value set is defined for the Δ parameter. Typical choices are the ranges $[1, 100]$ or $[1, 150]$. If (21) is used

$$\begin{pmatrix} \frac{p_1}{U_1}+\frac{p_2}{U_2} & -\frac{p_2}{U_2} & 0 & 0 & \dots & 0 & 0 & 0 \\ -\frac{p_2}{U_2} & \frac{p_2}{U_2}+\frac{p_3}{U_3} & -\frac{p_3}{U_3} & 0 & \dots & 0 & 0 & 0 \\ 0 & -\frac{p_3}{U_3} & \frac{p_3}{U_3}+\frac{p_4}{U_4} & -\frac{p_4}{U_4} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & -\frac{p_{NoD-2}}{U_{NoD-2}} & \frac{p_{NoD-2}}{U_{NoD-2}}+\frac{p_{NoD-1}}{U_{NoD-1}} \end{pmatrix} \cdot \begin{pmatrix} G(d_1) \\ G(d_2) \\ G(d_3) \\ \vdots \\ G(d_{NoD-2}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \frac{p_{NoD-1}}{UNoD-1} G(d_{NoD-1}) \end{pmatrix} \tag{23}$$

to set the disks' velocities, define a value set for the U parameter as well. Typical values are $[1, 10000]$.

2) In a similar manner, a range of values is selected for the NoD parameter, typically $[2, 10]$.
3) A set of values is also defined for the φ parameter of the disk workloads' distribution, typically 0.0 (uniform distribution) to 10.0 with steps of 0.1 units.
   In each of the above cases much larger value sets may be used with trivial processing time impact. Actual results though have indicated the use of values beyond the aforementioned ranges to be excessive.
4) For every possible set of values $(U, \Delta, NoD, \varphi)$ the linear system of every analysis' variation is constructed and solved, and through (10) the projected mean waiting time $\overline{D}$ of the server is calculated.

The set $(U, \Delta, NoD, \varphi, \text{Analysis-Variation})$ achieving the minimum $\overline{D}$ is considered to optimize the server's performance. The corresponding disk speeds U_i are calculated through either (20) or (21). The respective $G(d_i)$ are also reduced to their corresponding d_i. Thus the optimal disk sizes are set as well and the broadcast schedule is constructed as described in Section II-B.

Attention must also be paid to the fact that the value of the parameter L is subjected each time to the current disk speeds and sizes as implied in Section II-B and thus must be calculated accordingly when needed, as described by (1).

Finally, it must be stated that the procedure's execution time obviously depends on the value ranges of the $(U, \Delta, NoD, \varphi)$ parameters, as described in steps 1-3. These ranges define how many times the linear systems of step 4 will be constructed and solved. The computational complexity of steps 1-3 is $O_1 = O(|U| \cdot |\Delta| \cdot |NoD| \cdot |\varphi|)$, where $|.|$ represents the number of elements in the value set of the corresponding parameter. The computational complexity of step 4 corresponds to the construction and solution of a tridiagonal linear system for every analysis variant, and as such is $O_2 = 4 \cdot O(NoD)$ [25]. As a results, the complexity of the proposed OBSP scheme is $O = O_1 \cdot O_2$. Notice that the alternative [16, 19] would be to run O_1 whole system simulations, a task which is bound to be much more time consuming compared to O_2.

IV. Simulations and Results

A. Configuration

The main goal of the simulation was the validation of the client's mean waiting time produced by the OBSP for every major client case. The term client case refers to any different combination of the $Range$, $RegionSize$ and θ parameters, as it describes a unique client probabilistic model. Amongst them, the $Range$ parameter is subjected to the total amount of available server pages ($DBSize$). However, since the preceding papers [5, 16] set this parameter to the value 5000, the same convention will be used in the context of this work as well. The value sets of the $RangeRegionSize$ and θ parameters are given in Fig. 4-7 in the corresponding figure captions. The values for the $Range$ and $RegionSize$ parameters are chosen in tandem for obvious reasons.

Concerning the θ parameter, values smaller than one unit correspond to high degree of correlation between regions, while values in the range $(1, \infty)$ suggest the opposite, as stated in Section II-C. The values 0.5 and 1.5 represent the two corresponding extremes. The value $\theta = 1$ is not allowed due to a zipf p.d.f. restriction. Finally, the value 0.95 is the most commonly used in bibliography [5, 16].

For each of the aforementioned client cases, OBSP's performance is compared with the most outstanding alternatives, namely the GBSP and the CWDB procedure. Since the former employs (20) to set the disks' velocities while the later uses (21), two variants of the OBSP must be examined in order to promote the comparison's credibility:

1) The OBSP-Δ variant which employs (20) and is thus more fit to be compared with the GBSP, and
2) the OBSP-U variant which employs (21) and is therefore more eligible for comparison with the CWDB.

The values' sets of the parameters of each procedure are given in Table I.

The server is assumed to have perfect knowledge of the client's p.d.f as depicted in Fig. 2(a). For an amount of $30,000$ client queries the client's mean waiting time is observed and its mean value is calculated. In order to be in full compliance with [5] where the Broadcast Disks method was introduced, the client's $ThinkTime$ is taken into account and is set equal to 2 timeslots. In other words, upon receiving a requested page the client remains inactive for a period of 2 timeslots. very distinct simulation was repeated 100 times in order to extract a reliable mean value of the clients' mean waiting time.

Finally, it must be stated that through the preceding mathematical analysis, OBSP was able to pinpoint the optimal values for its parameters in a short time, through the use of (10) which projects the client's mean waiting time. These projected values are presented as well for each client case. On the other hand, not having a corresponding mathematical background to support such a task, GBSP and CWDB require a simulation to be run for every possible combination of the values of the parameters that comprise their corresponding value sets, an extremely time-consuming task.

TABLE I
VALUES' SETS OF THE PROCEDURES' PARAMETERS (UNARY STEP IS ASSUMED WHERE OMITTED)

Parameter name	Value Sets			
	GBSP	CWDB	OBSP	
			OBSP-Δ	OBSP-U
Δ	$[1, 100]$	$[1, 100]$	$[1, 100]$	$[1, 100]$
NoD	$[1, 10]$	$[1, 10]$	$[1, 10]$	$[1, 10]$
U	N/A	$[1, 10000]$	N/A	$[1, 1000]$
φ	N/A	N/A	$1 : 0.1 : 10$	$1 : 0.1 : 10$

B. Analysis and Simulation Results

The results are depicted in Fig. 4-7, where OBSP, CWDB and GBSP are compared performance-wise. In the case of OBSP both the estimated and the simulated values are presented, for both of the variants OBSP-Δ and OBSP-U. The goal is to:

1) Compare OBSP-U with CWDB performance-wise.
2) Compare OBSP-Δ with GBSP also performance-wise.
3) Compare the analytically projected values of OBSP-U and OBSP-Δ with their corresponding simulation-derived counterparts.
4) Compare OBSP with CWDB and GBSP overall.

Each algorithm's performance corresponds to the mean client waiting time achieved by it, and is measured in "time units". One time unit corresponds to the time interval required to broadcast one single page. It is reminded that all pages are of equal size.

C. Remarks

Concerning the performance of GBSP and the OBSP-Δ variant, the results have shown the latter to be dominant in all of the examined cases. The behavior of OBSP in general was uniform and somewhat expected, while GBSP's performance was in several cases erratic, as depicted in Fig. 4. In contrast with all other algorithms, the performance of GBSP suddenly rises and then begins to decline. Examination of the corresponding optimal parameters showed this phenomenon to occur when three-disk configurations begin to outperform the simple two-disk ones. The explanation for this phenomenon lies in the effect of the zipfian θ parameter on the client's p.d.f. which is depicted in Fig. 2(b). Low θ values produce a more uniform probability distribution. In this case the optimal solution is obvious: split the pages into two disks, one containing the useful pages and one containing the useless ones. GBSP successfully detects this fact, and thus both GBSP and OBSP-U behave similarly for $\theta \in [0.5, 0.8]$. From that point and on though, the distribution's aberration from uniformity becomes more and more dominant, and thus more refined, multi-disk configurations must be employed. Not relying on a strict mathematical analysis, GBSP fails to detect the optimal disk sizes, and thus a sudden increase in waiting time is observed which then follows the common declining scheme of all other cases.

From the comparison between CWDB and the OBSP-U variant, it is clear that the OBSP continues to outperform its rival in all examined cases. The K-means based grouping of pages into disks performed by CWDB does perform satisfactory to a degree, as was observed in [19] as well. However, the grouping process - as the mathematical analysis indicates - is closely bound to the selection of the disks' velocities. CWDB's page grouping is completely uncorrelated from the velocities' selection task. OBSP on the other hand, takes this fact into consideration through the mathematical analysis, and is thus enabled to perform better.

OBSP, and more specifically OBSP-U, outperforms all alternative approaches. Concerning the optimal accompanying parameters for each case, the ascending disk workload distribution was found to be dominant, with the steepness factor φ usually being equal to 0.5 or 1.0 units. Uniform distributions ($\varphi = 0$) also made their appearance sporadically. The DRG-Variant was also dominant in the vast majority of the cases. For high values of the $Range$ and θ parameters, mainly the NDRG but also the NDPG-Variants were proven superior. This is due to the fact that in these cases, inserting some useful pages in the last and slowest rotating disk forces the system to produce smaller optimal broadcast schedules in order for these pages to be broadcasted as more frequently as possible. Thus the

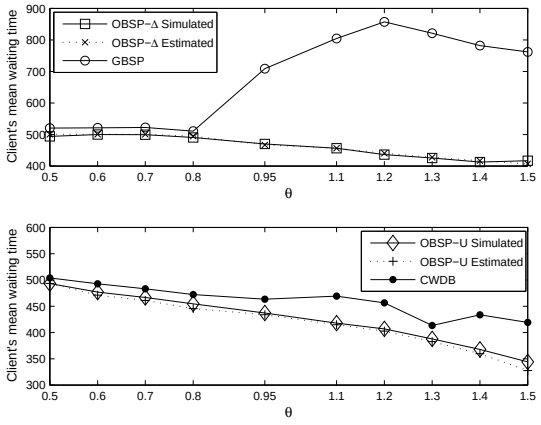


Fig. 4. Client's mean waiting time as a function of θ parameter for $Range = 1000$ and $RegionSize = 50$

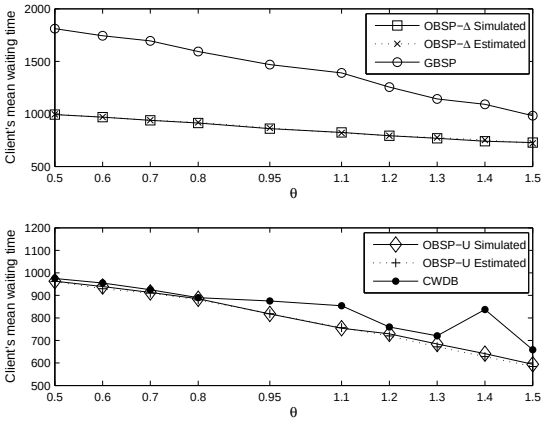


Fig. 5. Client's mean waiting time as a function of θ parameter for $Range = 2000$ and $RegionSize = 50$

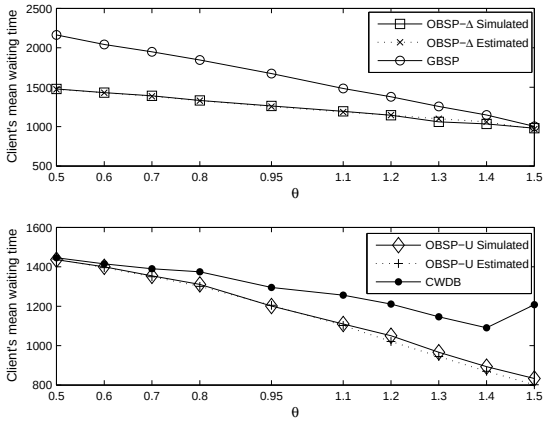


Fig. 6. Client's mean waiting time as a function of θ parameter for $Range = 3000$ and $RegionSize = 50$

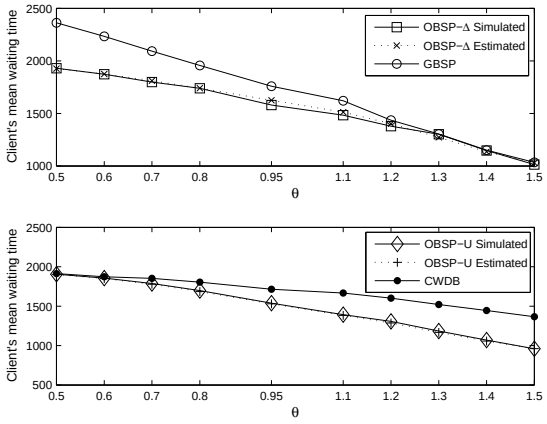


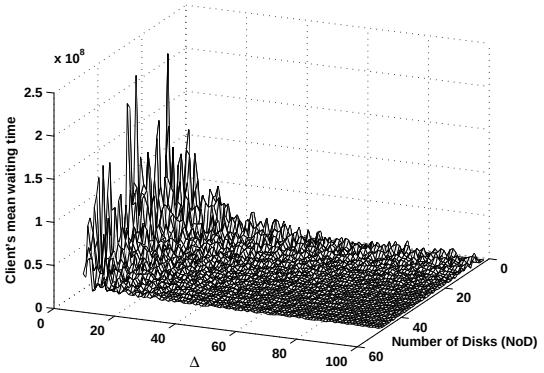
Fig. 7. Client's mean waiting time as a function of θ parameter for $Range = 4000$ and $RegionSize = 50$

clients' mean waiting time decreases as expected.

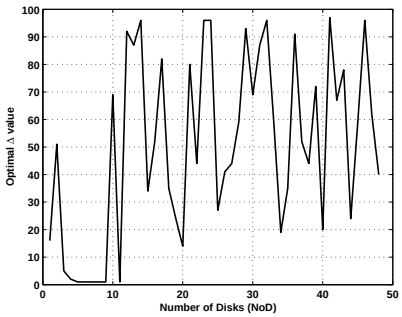
Another observation is that all algorithms tend to perform better as the zipfian θ increases. The explanation for this phenomenon lies on the effect of the θ parameter on the client's p.d.f. demonstrated in Fig. 2(b). High values of θ mean that a group of pages becomes increasingly more popular than the others. To compensate, all algorithms group this pages into increasingly smaller, faster rotating disks which obviously is in favor of the overall performance.

The values of the performance obtained through simulation and the mathematical analysis are almost in full accordance. Slight variations are attributed to the fortuity factors implicated in the simulations.

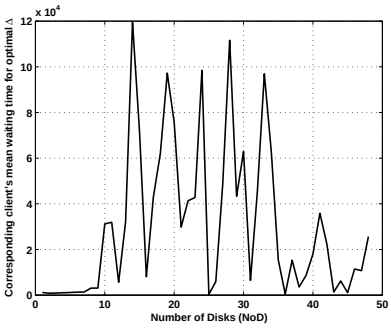
Finally, it became evident through the initial experimentation with the system's parameters that slight deviations from their optimal values could make or break its performance in an unpredictable way. In order to investigate this phenomenon more thoroughly, several random client cases were tested over a wide range of parameters' combinations and for every one of the GBSP, CWDB, OBSP-Δ and OBDP-U algorithms. In the end all yielded results identical to those



(a) The client's mean waiting time as a function of the number of disks (NoD) and Δ



(b) The optimal Δ value per NoD value



(c) The client's mean waiting time for each optimal Δ value, per corresponding NoD value

Fig. 8. Illustrations of the system's erratic behavior.

of Fig. 8. In Fig.8(a) the performance of the system for $\{DBSize = 5000, Range = 1000, \theta = 0.95, RegionSize = 50, algorithm =$"OBSP-$\Delta$"$\}$ and $NoD \in [2, 50] \times \Delta \in [1, 100]$ is illustrated. While performance tends to improve with the adoption of multi-disk, fast-rotating schemes, it nonetheless presents great fluctuations. In Fig.8(b), for each of the 50 NoD values, the Δ achieving the best performance is recorded and shown. The performance itself for these cases is illustrated in Fig.8(c).

It is readily obvious that no easily comprehendible relation connects the systems parameters with the systems' performance. Moreover, slight deviation of the parameters from their optimal values, as in the case of Δ in Fig.8(c), may downgrade the system's performance. It is thus imperative that

these parameters be defined through analytical tools.

V. Conclusion

Optimizing the performance of wireless broadcast-based systems is not a trivial task. The goal is to construct a common broadcast schedule that best fits the needs of a wide variety of clients. To this end, the Broadcast Disks method constitutes a popular broadcast schedule construction framework. This paper presented a mathematical analysis of this method and defined an analytical procedure aiming at minimizing the clients' mean waiting time. Comparison with other approaches yielded not only better performance in every one of the client cases, but faster processing times as well, since the optimal system parameters were pinpointed through the analysis rather than heuristically. Finally, it was shown that due to the system's erratic nature, one can define the optimal parameters' values in a credible way only through analytical methods.

References

[1] A. Bar-Noy, R. Bhatia, J. Naor, and B. Schieberaruch, "Minimizing service and operation costs of periodic scheduling," in *Proc. of the 9th SODA'98*, pp. 11–20.

[2] V. Gondhalekar, R. Jain, and J. Werth, "Scheduling on airdisks: Efficient access to personalized informationservices via periodic wireless data broadcast," in *Proc. IEEE Int. Conf. on Communications*, 1997, pp. 1276–1280.

[3] C. Kenyon and N. Schabanel, "The data broadcast problem with non-uniform transmission times," in *Proc. of the 10th SODA'99*.

[4] C.-L. Hu and M.-S. Chen, "Online scheduling sequential objects with periodicity for dynamic information dissemination," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 2, pp. 273–286, 2009.

[5] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: data management for asymmetric communication environments," in *Proc. of the SIGMOD'95*, pp. 199–210.

[6] S. Acharya, M. Franklin, and S. Zdonik, "Prefetching from broadcast disks," in *Proc. of the 12th ICDE'96*, pp. 276–285.

[7] K.-L. Wu, P. Yu, and M.-S. Chen, "Energy-efficient caching for wireless mobile computing," in *Proc. of the 12th ICDE'96*, pp. 336–343.

[8] M.-S. Chen, K.-L. Wu, K.-L. , and P. Yu, "Optimizing index allocation for sequential data broadcasting in wireless mobile computing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 1, pp. 161–173, 2003.

[9] J. Xu, W.-C. Lee, X. Tanga, Q. Gao, and S. Li, "An error-resilient and tunable distributed indexing scheme for wireless data broadcast," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 392–404, 2006.

[10] S. Acharya, M. Franklin, and S. Zdonik, "Balancing push and pull for data broadcast," in *Proc. of the SIGMOD'97*, pp. 183–194.

[11] C.-L. Hu and M.-S. Chen, "Adaptive multi-channel data dissemination: Support of dynamic traffic awareness and push-pull time balance," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 2, pp. 673 – 686, 2005.

[12] C.-J. Su, L. Tassiulas, and V. Tsotras, "Broadcast scheduling for information distribution," *Wireless Networks*, vol. 5, no. 2, pp. 137–147, 1999.

[13] Y.-I. Chang, C.-N. Yang, and J.-H. Shen, "A binary-number-based approach to data broadcasting in wireless information systems," in *Proc. of the IEEE WirelessCom'05*, pp. 348–353.

[14] P. Nicopolitidis, G. Papadimitriou, and A. Pomportsis, "Multiple-antenna data broadcasting for environments with locality of demand," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2807 – 2816, 2007.

[15] P. Nicopolitidis, G. Papadimitriou, and A. Pomportsis, "Using learning automata for adaptive push-based data broadcasting in asymmetric wireless environments," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, pp. 1652–1660, 2002.

[16] W. Yee, S. Navathe, E. Omiecinski, and C. Jermaine, "Bridging the gap between response time and energy-efficiency in broadcast schedule design," in *Proc. of the 8th EDBT'02*, pp. 572–589.

[17] W.-C. Peng and M.-S. Chen, "Dynamic generation of data broadcasting programs for a broadcast disk array in a mobile computing environment," in *Proc. of the 9th CIKM'00*, pp. 38–45.

[18] A. Konig and G. Weikum, "Combining histograms and parametric curve fitting for feedback-driven query result-size estimation," in *Proc. of the 25th VLDB'99*, pp. 423–434.

[19] C. Liaskos, S. Petridou, G. Papadimitriou, P. Nicopolitidis, M. Obaidat, and A. Pomportsis, "Clustering-driven wireless data broadcasting," *IEEE Wireless Communications Magazine*, to appear.

[20] C. Liaskos, S. Petridou, G. Papadimitriou, P. Nicopolitidis, and A. Pomportsis, "An analytical approach to the design of wireless broadcast disks systems," in *Proc. of the 9th ISADS'09*.

[21] Y. Saygin and O. Ulusoy, "Exploiting data mining techniques for broadcasting data in mobile computing environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 6, pp. 1387–1399, 2002.

[22] S. Acharya, M. Franklin, and S. Zdonik, "Dissemination-based data delivery using broadcast disks," *IEEE Wireless Communications*, vol. 2, no. 6, pp. 50–60, 1995.

[23] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, 1975.

[24] L. Pietronero, E. Tosatti, V. Tosatti, and A. Vespignani, "Explaining the uneven distribution of numbers in nature: The laws of benford and zipf," *Physica A*, pp. 297–304, 2001.

[25] G. Golub and C. V. Loan, *Matrix Computations*. London: John Hopkins University Press (3rd Edition), 1996.

Christos K. Liaskos received the Diploma and Ph.D. degrees in Computer Science from the Aristotle University of Thessaloniki, Greece in 2001 and 2007 respectively. He is currently an adjunct lecturer at the University of Western Macedonia, Greece. His research interests include optical networks and optical switching.

Sophia G. Petridou ($M'08$) received the Diploma and Ph.D. degrees in Computer Science from the Aristotle University of Thessaloniki, Greece in 2000 and 2008 respectively. Her research interests include clustering, optical and wireless networks.

Georgios I. Papadimitriou ($M'89$, $SM'02$) received the Diploma and Ph.D. degrees in Computer Engineering from the University of Patras, Greece in 1989 and 1994 respectively. In 1997 he joined the faculty of the Department of Informatics, Aristotle University of Thessaloniki, Greece, where he is currently serving as an Associate Professor. His main research interests include optical networks and wireless networks. Prof. Papadimitriou is Associate Editor of five IEEE Journals. He is co-author of three books published by Wiley. He is author or coauthor of 75 journal and 85 conference papers. He is a Senior Member of the IEEE.

Petros Nicopolitidis received the B.S. and Ph.D. degrees in Computer Science from the Department of Informatics, Aristotle University of Thessaloniki, in 1998 and 2002, respectively. Since 2004 he is a Lecturer at the same Department. His research interests are in the areas of wireless networks and mobile communications. He is coauthor of the book Wireless Networks (New York: Wiley, 2003).

Andreas S. Pomportsis received the B.S. degree in physics and the M.S. degree in electronics and communications, both from the University of Thessaloniki, Greece, and the Diploma in electrical engineering from the Technical University of Thessaloniki, Greece. In 1987, he received the Ph.D. degree in computer science from the University of Thessaloniki. Currently, he is a Professor at the Department of Informatics, Aristotle University, Thessaloniki, Greece. He is coauthor of the books "Optical Switching" (Wiley, 2007) and "Multiwavelength Optical LANs" (Wiley, 2003) and "Optical Networks" (Wiley, 2003). His research interests include computer networks, computer architecture, parallel and distributed computer systems, and multimedia systems.