A Novel Clustering-Driven Approach to Wireless Data Broadcasting

C.K.Liaskos[1], S.G.Petridou[1], *Member, IEEE,* G.I.Papadimitriou[1], *Senior Member, IEEE,* P.Nicopolitidis[1], M.S.Obaidat[2], *Fellow, IEEE,* A. S. Pomportsis[1]
[1]Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece
[2]Department of Computer Science, Monmouth University, West Long Branch, NJ 07764

A new approach to the design of wireless data broadcasting schemes is introduced. The proposed clustering-driven wireless broadcasting procedure embellishes the popular broadcast disks model, with a clustering mechanism which significantly improves its performance. The proposed scheme is compared with other classical procedures and is shown to be dominant in nearly every possible client configuration case. A thorough performance study leads to important conclusions regarding the impact of the various steps of the broadcast disks method on the performance of the system and indicates the superiority of the proposed clustering-driven approach.

I. Introduction

The performance of wireless push based communication systems [1] heavily depends on the proper scheduling of the data items' broadcast. It is commonly agreed that every broadcast must present periodicity and proportionality: the interval between two consecutive appearances of the same data item in the schedule must be constant, and the total number of its appearances must be proportional to its popularity. The broadcast disks method [2] was developed to provide a lax-yet compliant with the aforementioned criteria-framework. Every instantiation of this method consists of four steps: Firstly, a feedback mechanism provides a measure of each data item's popularity. A grouping algorithm then organizes the items in collections called Disks, according to their popularity. These represent an array of physical disks, spinning around a common axis. Each of these disks is then set to spin with an angular velocity proportional to the aggregate demand of its contained pages. 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.

Purpose of this paper is to present a novel instantiation of the broadcast disks method, the clustering-driven wireless data broadcasting procedure (CWDB), and thoroughly compare it through simulation with another classical solution, the GREEDY [3] based broadcasting procedure (GBBP); both are named after the grouping algorithm they incorporate. To our knowledge, this is the first time that not only complete instances of the Broadcast Disks method are presented and tested, but the comparison concerns such a variety of different client configurations as well.

The remainder of this paper is organized as follows. Section II provides the network configuration and operation, including the new feedback mechanism. In section III the CWDB procedure is presented, alongside GBBP. Detailed information on the simulation configuration is given in Section IV, accompanied by the produced results in each case. Conclusions are given in Section V.

II. Network Operation and Feedback Mechanism

The physical network layout assumed in this paper is typical for all push-based systems and consists of a database and broadcast scheduling server, a broadcast mechanism and a number of wireless clients. The database contains a number of *DBSize* equally sized items, organized in equally-sized groups called ***Regions.*** Any client is considered to access only a random subset of the total server pages, and their number will be denoted as *Range*. All *RegionSize* pages in one Region have equal probability of being requested by the client. The Regions themselves follow the zipf probability distribution. Thus if we sort all Regions according to their request probability in a descending fashion, the request probability of any page in region r_i will be

$$P(r_j) = \frac{c}{j^{\theta}},\ j = 1..\left\lceil \frac{\text{Range}}{\text{RegionSize}} \right\rceil \qquad (1)$$

$\boldsymbol{\theta}$ being the zipf p.d.f. parameter (and as such $\theta \neq 1$) and $\boldsymbol{c}$ a constant value conforming to the condition

$$\sum P(r_j) = 1 \qquad (2)$$

The network operation is a repeated procedure and is depicted in fig. 1. At any given moment, if the currently broadcasted page satisfies the needs of a client, he replies with a single approval message (e.g. a single binary digit in a noiseless environment) [4]. Thus the server acquires a posteriori knowledge about the clients' demands, with a minimal client upload rate. The server maintains a *Votes registry*, which holds the aggregated votes for each page in the server database. After a valid period of time the voting stops and the server uses a grouping, a disk speed definition and a broadcast schedule constructor algorithm to produce the new broadcast sequence as already described. The registry is then nullified and the new broadcast commences.

III. The Compared Procedures

The proposed CWDB and the well-known GBBP are summarized in table 1. Both procedures utilize the same feedback mechanism and broadcast scheduling scheme but different grouping and disk speed definition algorithms. GBBP uses the GREEDY grouping algorithm on the completed Votes registry in order to produce the new Disks. It then sets their speeds as the terms of an arithmetic progression where the final term is equal to one unit and corresponds to the last and less popular Disk. This is in accordance

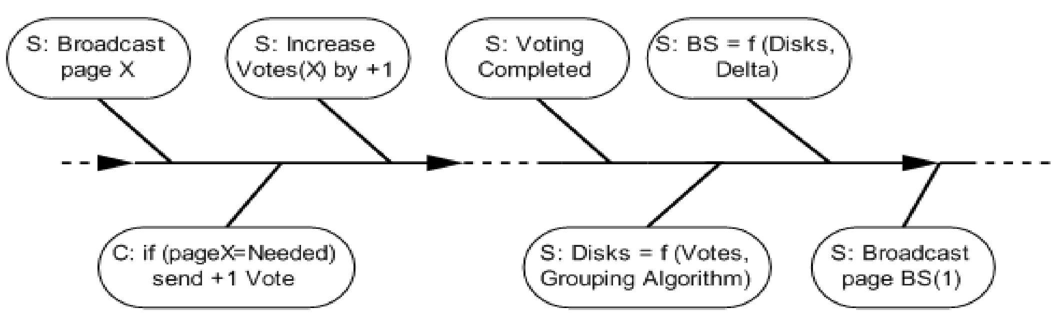

Fig.1 Network operation and feedback mechanism – "S" denotes server activity, while "C" stand for a client's action

Table 1. CWDB and GBDB comparison

	GBBP	CWDB
Provide Feedback	*Complete the Votes registry, as described in Section II*	*Complete the Votes registry, as described in Section II*
Perform Data Grouping	Set N_{Disks}=N Disk(1..N_{Disks})=GREEDY(Votes, N_{Disks})	Set N_{Disks}=N Foreach page "p" in Votes do If (Votes(p) equals 0) Insert page "p" into last Disk end end set Clusters=N_{Disks}-1 Disk(1..N_{Disks}-1)=KMEANS(Votes,Clusters)
Set Disk Speeds	$U_i=(N_{Disks}-i)\Delta+1$	Set $U_{NDisks}=1$ and $U_{NDisks-1}$=*Custom Value* Set $U_i=U_{NDisks-1}[(N_{Disks}-1-i)\Delta+1]$, i=1..($N_{Disks}$-2)
Schedule Broadcast	*The default scheduling algorithm of the Broadcast Disks method*	*The default scheduling algorithm of the Broadcast Disks method*

with [3] where GREEDY was firstly presented. CWDB on the other hand, first checks for any useless (zero-voted) pages and groups them as the last Disk, whose speed is set to one unit. The K-means [5] clustering algorithm is then applied to the remaining entries of the Votes registry and the rest of the disks are formed. The speed of the second-to-last Disk ($U_{NDisks-1}$) represents the difference in importance between the useful and the useless pages and is manually set to a proper value. The remaining velocities follow the arithmetic progression rule.

IV. Simulation Results

Both procedures were simulated under a wide range of values for nearly every parameter of the system. Their exact value sets are given in table 2. Each possible combination of the first two parameters of this table represents a unique client configuration. The following three parameters define the mean response time per client query for each case. For each case we keep the smallest achieved mean response time and the corresponding parameters. Some indicative results are presented analytically in table 3 and graphically in fig. 2-5.

The following conclusions can be derived from the simulation results:

i. The novel CWDB outperforms GBBP in the vast majority of the test cases with the difference of performance increasing as ClientRange decreases. This was expected as CWDB relies on the existence of a relatively big amount of unused pages.

ii. In the cases were both CWDB and GBBP produce similar disk configurations, the difference in performance can only be ascribed to the new speed velocity definition algorithm firstly introduced in this paper.

iii. The optimal number of disks is usually two, three or four. The three-disk configurations are the most common. Values higher than four are extremely rare.

Table 2. Simulation configuration value sets

Parameter Name	Value Set
θ (zipf p.d.f)	0.3 0.5 0.7 0.95 1.1 1.5
Range/RegionSize	30/1 30/3 30/5 1000/30 1000/50 2000/50 2000/100 3000/50 4000/50
Δ	1 2 3 4 5 6 7 8 15 20 50 80 100
Number of Disks (N)	2 3 4 5 6 7
U_{N-1} (CWDB only)	10 30 50 100
DBSize	5000

Table 3.Indicative simulation results.

θ		Range /RegionSize	30/5	1000/30	1000/50	2000/100
0.3	CWDB	**Response time** **(Δ, U_{N-1})** **[Disk Sizes]**	**29** (3,100) [18 12 4970]	**517** (5,100) [1000 4000]	**518** (4,100) [1000 4000]	**1025** (1,100) [2000 3000]
	GBBP	**Response time** **Δ** **[Disks Sizes]**	**89** 1 [13 17 4970]	**666** 15 [990 4010]	**695** 50 [996 4004]	**1870** 1 [1562 3474]
0.7	CWDB	**Response time** **(Δ, U_{N-1})** **[Disk Sizes]**	**35** (2,100) [6 24 4970]	**471** (1,100) [30 157 813 4000]	**480** (1,100) [50 184 766 4000]	**995** (1,100) [250 1749 3001]
	GBBP	**Response time** **Δ** **[Disk Sizes]**	**89** 50 [12 18 4970]	**766** 6 [331 618 4051]	**712** 15 [347 620 4033]	**1778** 1 [44 74 114 294 4474]
0.95	CWDB	**Response time** **(Δ, U_{N-1})** **[Disk Sizes]**	**33** (3,100) [8 22 4970]	**430** (1,100) [28 92 876 4004]	**439** (1,100) [48 155 797 4000]	**870** (1,100) [98 300 1601 3001]
	GBBP	**Response time** **Δ** **[Disk Sizes]**	**91** 50 [11 19 4970]	**839** 5 [87 165 606 4142]	**845** 6 [275 602 4123]	**1607** 1 [96 93 223 724 3864]
1.5	CWDB	**Response time** **(Δ, U_{N-1})** **[Disk Sizes]**	**30** (2,100) [3 5 22 4970]	**364** (3,30) [28 60 902 4010]	**356** (2,100) [49 92 857 4002]	**902** (2,30) [98 191 1623 3088]
	GBBP	**Response time** **Δ** **[Disk Sizes]**	**95** 50 [8 22 4970]	**720** 7 [106 390 4504]	**779** 6 [139 428 4433]	**1145** 3 [84 124 543 4249]

iv. The k-means clustering algorithm employed by CWDB tends to perform a more selective grouping of pages than GBBP. The faster disks produced by CWDB usually contain less pages than their GBBP counterparts.

v. In all but few cases CWDB produces the best response time for Δ=1 and U_{N-1}=100. Thus there is no pressing need to create an optimizing scheme regarding these parameters. On the other hand, the optimal Δ for GBBP ranges from 1 to 50, making the need for such a scheme imperative.

vi. It is clear that the ClientRange parameter greatly affects the system's response time. Thus it is made obvious that every broadcast scheduling procedure must be tested against a variety of client configurations in order to evaluate its performance in a realistic manner.

vii. The θ parameter affects the response time indirectly by defining the optimal duration of the voting period. This is why its effect is trivial for small client ranges but increases for bigger values of this parameter. The actual computation of the optimal duration of the voting period is a work in progress. A more realistic, multi-client simulation model is also under development.

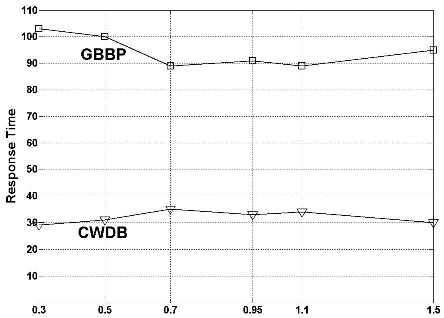

Fig.2. CWDB vs GBBP for Range=30, RegionSize=5

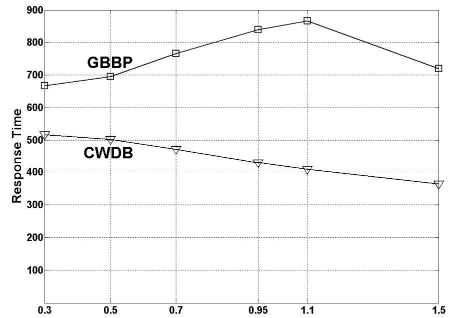

Fig.3. CWDB vs GBBP for Range=1000, RegionSize=30

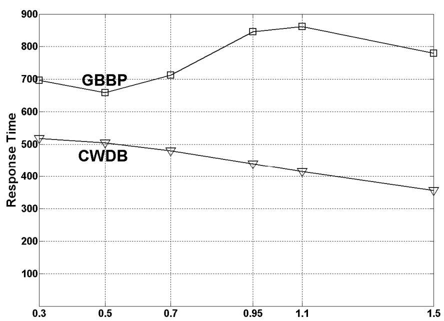

Fig.4. CWDB vs GBBP for Range=1000, RegionSize=50

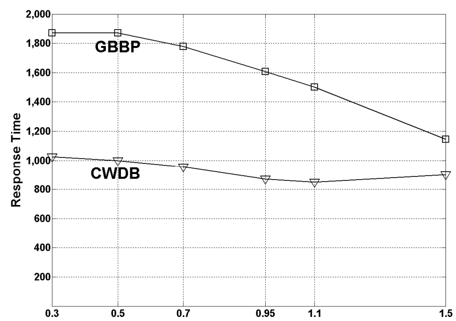

Fig.5. CWDB vs GBBP for Range=2000, RegionSize=100

Conclusion

In the context of this paper, a new clustering-driven wireless data broadcasting procedure has been presented and tested thoroughly. Performance comparison with other classical solutions suggested that the use of clustering algorithms can be the basis of a new generation of high performance data broadcasting schemes.

References

[1] P.Nicopolitidis, G.I.Papadimitriou, M.S.Obaidat and A.S.Pomportsis, "*Performance Optimization of an Adaptive Wireless Push System in Environments with Locality of Demand*", Computer communications, Elsevier,vol.29, no.13-14, August 2006, pp.2542-2549.

[2] Swarup Acharya, Rafael Alfonso, Michael Franklin, Stanley Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments", *Proceedings of the ACM SIGMODConference*, San Jose, CA, May 1995

[3] Wai Gen Yee, Shamkant B. Navathe, Edward Omiecinski, Christopher Jermaine, "Bridging the Gap between Response Time and Energy-Efficiency in Broadcast Schedule Design", *Proceedings of the 8th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 572–589, 2002

[4] Petros Nicopolitidis, Georgios I. Papadimitriou, Andreas S. Pomportsis, "Using Learning Automata for Adaptive Push-Based Data Broadcasting in Asymmetric Wireless Environments", *IEEE Transactions on vehicular technology*, vol. 51, no. 6, November 2002

[5] S. Petridou, P. Sarigiannidis, G. Papadimitriou and A. Pomportsis, "On the use of Clustering Algorithms for Message Scheduling in WDM Star Networks", *IEEE Journal of Lightwave Technology*, to appear, 2008.