More for Less
Getting More Clients by Broadcasting Less Data

Christos Liaskos[1], Ageliki Tsioliaridou[2], and Georgios I. Papadimitriou[1,⋆]

[1] Dept. of Informatics, Aristotle University, Thessaloniki 54124, Greece
{cliaskos,gp}@ee.auth.gr
[2] Dept. of Electrical and Computer Engineering, Democritus University, Xanthi 67100, Greece
atsiolia@ee.duth.gr

Abstract. Broadcasting is scalable in terms of served users but not in terms of served data volume. Additionally, waiting time deadlines may be difficult to uphold due to the data clutter, forcing the clients to flee the system. This work proposes a way of selecting subsets of the original data that ensure near-optimal service ratio. The proposed technique relies on the novel *data compatibility distance,* which is introduced herein. Clustering techniques are then used for defining optimal data subsets. Comparison with related work and brute force-derived solutions yielded superior and near-optimal service ratios in all test cases. Thus, it is demonstrated that a system can attract more clients by using just a small portion of the available data pool.

Keywords: periodic broadcasting, service ratio, content selection.

1 Introduction

Data broadcasting is an efficient means of bandwidth preservation and information advertising. As the Internet expands incorporating a steadily increasing number of entities, common needs and preferences begin to appear among the clients. This fact calls for dissemination of information per user class instead of per single user, thus saving network resources by employing broadcasting. However, it is typical of contemporary content providers to attempt coverage of all information topics, in an effort to attract as many clients as possible. Nevertheless, broadcasting does not scale well when the data volume increases [11]. Users experience too long waiting times and flee the system. The present study addresses this issue in the context of wireless, push-based broadcasting.

Push-based broadcasting [1] relies solely on the knowledge of the data popularity distribution, disregarding individual client queries. Its opposite, pull-based process [7], deals with individual, known client queries, serializing then in an optimal manner. In terms of system setup, wireless broadcasting typically assumes

⋆ This research has been co-financed by the European Social Fund - ESF and Greek national funds through the Research Funding Program "Heracleitus II" - NSRF.

a single frequency or cellular network, which covers a densely populated area. The clients therein are assumed to be interested in a common set of discrete data classes. Each class is associated with a request probability and an aggregate size of contained data measured in bytes. The class request probabilities may vary with time and any change is detected and handled by smart learning algorithms which are examined elsewhere, as a separate field of study [8, 14]. Once the class probabilities have become known, a central authority creates a broadcast schedule which optimizes a given criterion [2, 17]. The study of [4] proved that a periodic schedule minimizes the mean client waiting time. Periodicity refers to maintaining an approximately constant interval between consecutive broadcasts of each data class. The scheduling problem is then twofold: a) define the optimal number of occurrences of each class in the broadcast schedule and b) serialize data as periodicly as possible [11]. Once the final schedule has been constructed, it is broadcasted repeatedly over the wireless clients in range.

Initially, related studies assumed that a client may wait indefinitely for a wanted data class [4, 16, 17] and formulated the *square root rule*. The rule expresses the optimal number of occurrences for each class, in the form of a ratio of irrational numbers. However, these studies exhibited high computational complexity and subsequent works addressed the issue while still disregarding deadlines [10–12]. An initial attempt to reinstate deadlines took place in [6] where the authors presented a way of minimizing the variance of the clients' waiting time. Modified versions of the algorithms of [17] were demonstrated, that achieved tunable trade-off between the variance and the mean value of the waiting time, without discriminating between variance-sensitive and variance-independent applications. Subsequent studies model the clients' psyche, with a particular interest in impatience [3, 5, 15]. The optimization procedure follows the pattern of [17], typically applying the method of Lagrange multipliers, substituting the waiting time formula with a study-specific impatience metric. The impatience for a specific class is generally a rising function of the waiting time.

The problem with the related approaches is that they aim at improving the quality of the provided service, but not necessarily maximize the number of subscribed users. The issue stems from the fact that no data selectivity mechanism is offered. Even when presented with a huge bulk of obviously nonmatching data, the related approaches will attempt to disseminate all of them in a way that optimizes a given criterion. However, the very fact the volume of data has increased may hinder the upholding of the client deadlines. Furthermore, recent paradigms (e.g. Facebook) have demonstrated that a system becomes prosperous when the number of subscribed users is maximized, not necessarily requiring a top level of quality of service.

Differentiating from the majority of the related works, the present study will pursue to directly maximize the total number of the clients subscribed to the system. Initially, the optimal number of class occurrences inside the broadcast schedule will be defined. Notice that the second part of the scheduling process, i.e. a nearly-optimal serialization procedure, has been already proposed by the authors in [11]. Subsequently, it will be proven that optimality is not always possible for the complete data set. We will then establish a procedure that can select an optimizable data subset that yields a high number of system clients.

1.1 Notation and Standard Assumptions

We assume a set of data items organized in classes, which is to be disseminated to a group of wireless clients. The operation of the system is push-oriented and subscription-based; the clients do not post queries to a server, but rather listen to the broadcast stream, waiting for updates on classes of information that are of interest to them. However, their attention span is limited; as their waiting time increases, the probability of abandoning the system for another, better service increases as well. These parameters are modeled in the form of the following information class attributes:

- The class index, $i = 1 \ldots N$.
- The popularity, p_i, expressing the probability that a given client request refers to class i.
- The cardinality c_i of class i measured in bytes, which is equal to the total size of individual data items contained therein.

A client may not wait for a wanted item indefinitely. Related studies [5] model the clients' attention span as an exponential probability distribution,

$$P_{abandon}(w) = s \cdot e^{-s \cdot w} \tag{1}$$

which expresses the probability of a client waiting for time w before abandoning a query and, potentially, the system. The attribute s regulates the steepness of the distribution. High values correspond to more demanding clients.

The class popularity distribution p_i and criticality metric s can be inferred from user subscriptions [17], Facebook profile data, simple polling or automated, adaptive schemes [14]. The present study refers to the scheduling process that follows the estimation of these attributes.

The final schedule has a total size of:

$$C = \sum_{i=1}^{N} v_i \cdot c_i \tag{2}$$

while the interval between two consecutive appearances of class i in the schedule is equal to:

$$w_i^{max} = \frac{C}{v_i} \tag{3}$$

where v_i represents the number of occurrences of class i in the schedule. Finally, as in the totality of the cited works, the idealized wireless environment is used as a generic, broadcasting-affinitive context. The scheduling algorithms presented in this study can be applied to any other broadcast-based environment (e.g. wired multicasting). The proposed dissemination scheduling technique is generic enough to be agnostic of physical layer issues such as ray propagation, coverage, modulation type or encoding. In order to facilitate the presentation of the paper, the reader is encouraged to assume a TV broadcasting scenario, with the data items representing TV shows or classes of shows (e.g. politics, sports, news, etc.).

2 Analysis

Assume that there exist N available data classes with attributes $\{p_i, c_i\}$, $i = 1 \ldots N$. The well-known study of [5, Theorem 1] concluded that the class occurrences v_i^{opt} that maximize the client service ratio obey to the rule:

$$\frac{p_i}{c_i}\left[\frac{1}{C\cdot s}-\left(\frac{1}{C\cdot s}+\frac{1}{v_i^{opt}}\right)\cdot e^{-s\cdot\frac{C}{v_i^{opt}}}\right] = const.,\ i = 1\ldots N \tag{4}$$

The same study states that eq. (4) cannot be transformed further, and thus an analytic formula for v_i^{opt} is not provided. The authors then provide an algorithmic procedure that seeks to uphold the restrictions of (4) heuristically.

We begin by arguing that the actual problem is not the derivation of an analytic formula. In fact, such an expression is derived in the context of the work. What actually poses a serious issue is that the restrictions (4) cannot hold in most cases: Since v_i parameters express class occurrences in a schedule, they must always take positive integer values. However, equation (4) offers no such guarantee. In order to address this issue, we will be begin by extracting the analytic solution of (4) for v_i^{opt}.

Corollary 1. *The optimal class occurrence ratio, v_i^{opt}, that maximizes the client service ratio follows the relation:*

$$v_i^{opt} \propto \frac{-s}{1+W\left(\frac{-p_i+V\cdot c_i\cdot s}{e\cdot p_i}\right)} \tag{5}$$

where $W(.)$ is the Lambert-W function and V a constant which is calculated from the expression:

$$\sum_{i=1}^{N} v_i^{opt}\cdot c_i = 1 \tag{6}$$

Proof. Begin by transforming the ratio of (5) into real occurrences by multiplying both parts by the total schedule size, C. The proportionality then becomes equality which is solved for $W(x)$. The Lambert-W function, $W(x)$, represents the solution to $x = W\cdot e^W$, which after trivial calculations leads to the original equation (4). Finally, restriction (6) is derived from (2) when both parts are divided by C.

The class occurrences must be expressed as positive integers. However, the proportionality of equation (5) ensures that a simple rounding can be applied with

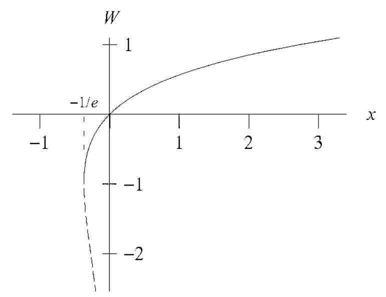


Fig. 1. A graphical illustration of the Lambert-W function, $W(x)$. The restriction $W \leq -1$ corresponds to $-1/e \leq x \leq 0$.

trivial precision loss, provided that the proportionality constant is big enough. Thus, the sole substantial constraint is:

$$v_i^{opt} \geq 0 \iff W\left(\frac{-p_i + V \cdot c_i \cdot s}{e \cdot p_i}\right) \leq -1, i = 1 \dots N \quad (7)$$

The Lambert-W function, $W(x)$, represents the solutions to $x = W \cdot e^W$ for a given x. It is a multivalued function, which assigns two W values to each $x \in (-1/e, 0]$. For $x = -1/e$ the function is single valued and equals $W(-1/e) = -1$. Figure 1 presents a graphical illustration of the function.

The restriction $W(x) \leq -1$ of equation (7) corresponds to $-1/e \leq x \leq 0$. Notice that the duality of the W-values in $[-1/e, 0]$ does not actually offer two alternatives. Should the upper branch of $W(x)$ be employed, the item broadcast frequencies would be negative, which has no physical meaning. Therefore, for all $i = 1 \dots N$ it must hold that:

$$-1/e \leq \frac{-p_i + V \cdot c_i \cdot s}{e \cdot p_i} \leq 0 \iff 0 \leq V \leq \frac{p_i}{c_i \cdot s}, \forall i \quad (8)$$

and equivalently:

$$0 \leq V \leq min\left\{\frac{p_i}{c_i \cdot s}\right\} \quad (9)$$

Remark 1. Recall from Corollary 1 that V is a constant which is defined through equation (6). In this context, relation (9) states that the value set of V is limited by the class with the smallest $\frac{p_i}{c_i \cdot s}$ ratio. In other words, if the optimization of equation (4) is unsolvable, then the class with minimal $\frac{p_i}{c_i \cdot s}$ ratio is to blame. In that sense, this class is incompatible with the others in terms of content.

The second condition of solvability stems from relation (6) of Corollary 1. The restriction can be rewritten in the form of the following equation:

$$f(V_o) = \sum_{i=1}^{N} v_i^{opt}(V_o) \cdot c_i - 1 = 0 \quad (10)$$

where $V_o \in [0, min\left\{\frac{p_i}{c_i \cdot s}\right\}]$, in accordance with (9).

Notice from Fig. 1 that the Lambert-W function is monotonous for the studied case of $W(V) < -1$. Therefore, the function

$$v_i^{opt}(V) = \frac{-s}{1 + W(V)} \tag{11}$$

is also monotonous, and so is the summation over all i. Thus, the function $f(V)$ of (10) is monotonous. Consequently, only a single value, $\{V_o | f(V_o) = 0\}$, may exist. Furthermore, the Bolzanno theorem must hold in the range $[0,\ min\left\{\frac{p_i}{c_i \cdot s}\right\}]$. According to it, the function must undergo a sign change:

$$f(0) \cdot f(min\left\{\frac{p_i}{c_i \cdot s}\right\}) \leq 0 \tag{12}$$

Theorem 1. *A set of information classes with attributes* $\{c_i,\, p_i\}$, $i = 1 \ldots N$ *is solvable in terms of maximizing the ratio of served clients, if it holds that:*

$$\sum_{i=1}^{N} \frac{s \cdot c_i}{1 + W\left[\frac{1}{e}\left(\frac{min\left\{\frac{p_i}{c_i \cdot s}\right\}}{\frac{p_i}{c_i \cdot s}} - 1\right)\right]} > -1 \tag{13}$$

Proof. Relation (13) comes as a direct expansion of (12).

Through relation (13), Theorem 1 provides a way of quickly checking the solvability of a data class set. Furthermore, Remark 1 can be used for pinpointing incompatible classes. Thus, an algorithm that detects solvable, optimal subsets of the original classes can be formulated.

2.1 Solvable Data Subsets that Yield Maximum Number of Clients

Let x_i denote the ratio:

$$x_i = \frac{min\left\{{}^{p_i}/_{c_i \cdot s}\right\}}{{}^{p_i}/_{c_i \cdot s}} \tag{14}$$

Equation (13) is transformed as:

$$\sum_{i=1}^{N} \frac{x_i \cdot p_i}{1 + W\left[\frac{1}{e}(x_i - 1)\right]} > -min\left\{{}^{p_i}/_{c_i \cdot s}\right\} \Longrightarrow$$

$$\left|\sum_{i=1}^{N} \frac{x_i \cdot p_i}{1 + W\left[\frac{1}{e}(x_i - 1)\right]}\right| < \left|min\left\{{}^{p_i}/_{c_i \cdot s}\right\}\right| \tag{15}$$

Consider the case of two data classes, $i = 1, 2$. Assume further that $x_2 = 1$, i.e. the second class has the lowest ${}^{p_i}/_{c_i \cdot s}$ value. From Fig. 1 notice that $W[0] \to -\infty$

in the studied case of $W < -1$. Therefore, in order for the two classes to be *compatible* (i.e. solvable), it must hold that:

$$\left|\frac{x_1 \cdot p_1}{1 + W\left[\frac{1}{e}\left(x_1 - 1\right)\right]}\right| < min\left\{{}^{p_i}/_{c_i \cdot s}\right\} \tag{16}$$

Relation (16) quantifies the compatibility of any two information classes. At first, the class with the minimum ${}^{p_i}/_{c_i \cdot s}$ ratio is detected. For the remaining class, the quantity $\mathcal{S} = \left|\frac{x_1 \cdot p_1}{1+W\left[\frac{1}{e}(x_1-1)\right]}\right|$ is calculated. If $\mathcal{S}$ is smaller than $\mathcal{R} = min\left\{{}^{p_i}/_{c_i \cdot s}\right\}$, the classes are compatible and may be broadcasted together in a way that maximizes the client service ratio. If not, the classes are incompatible. Therefore, we can define a *solvability distance*, which will measure the compatibility of two given classes:

$$\mathcal{D} = \begin{cases} \frac{\mathcal{S}}{\mathcal{R}-\mathcal{S}} & , 0 \leq \mathcal{S} < \mathcal{R} \\ \infty & , \mathcal{S} \geq \mathcal{R} \end{cases} \tag{17}$$

The idea is to use the metric of (17) in an iterative clustering approach. Such schemes, such as the classic K-means algorithm [9] typically operate as follows. At first, a predefined number of classes are randomly selected from the original set. These are called cluster centers or *centroids*. Then, the distance from every available class to each centroid is calculated. The classes are then assigned to the centroid that is nearest to them, thus forming groups. At this point new centroids are selected. These are set as the classes nearest to the center of the corresponding formed groups. The process is repeated iteratively, until no change has been made to the groups or to the centroids.

We proceed to formulate a novel procedure that employs this customized version of the k-means algorithm for data clustering. The updated k-means uses the metric of equation (17 - *solvability distance*) for calculating the distance between a centroid and any other class. The update procedure constitutes of selecting the class:

$$I = argmin_{(i)}\left\{\left|{}^{p_i}/_{c_i \cdot s} - E\left[{}^{p_i}/_{c_i \cdot s}\right]\right|\right\},\ i \in CurrentCluster \tag{18}$$

as the new centroid. $E[.]$ denotes the mean value of a set (.). This modified version is then incorporated to the novel, Data Clustering Algorithm for Higher Service Ratio (DCA-HSR).

The algorithm attempts at first to create as big as possible, solvable clusters of data classes. Thus, the NoC variable, which regulates the number of clusters, is initialized to 1 and is then increased by unary steps. The algorithm checks the solvability of every cluster created at each step. If a cluster is solvable, the algorithm calculates the *coverage* of a cluster as follows:

$$Coverage = \sum_{\forall i \in CurrentCluster} p_i \tag{19}$$

The coverage metric represents the maximum service ratio that the cluster may achieve. Notice that when a class is dropped from the scheduling process, the

Algorithm 1. Data Clustering Algorithm for Higher Service Ratio (DCA-HSR)

INPUT: A set of N CLASSES: $\{p_i, c_i\}$
OUTPUT: The cluster with the maximum service ratio, **best_cluster.**

```
best_coverage=0;
FOR NoC=1 to N, step 1
    [CLUSTERS]=ModifiedKmeans(CLASSES, NoC);
    FOREACH cluster in CLUSTERS
        IF (cluster is solvable) //eq (13)
            coverage=sum(cluster.p_i);
            IF (coverage>best_coverage)
                best_coverage=coverage;
                best_cluster=cluster;
            ENDIF
        ENDIF
    ENDFOR
ENDFOR
```

total service ratio is guaranteed to decrease by the corresponding request probability, p_i. The algorithm proceeds to select the solvable cluster that yields the maximum coverage.

Since the selected cluster is guaranteed to be solvable, equation (5) can produce the optimal number of periodic class occurrences in the broadcast schedule, v_i^{opt}. These values, alongside the class cardinality attributes, c_i, are then passed as input to the serializing process of [11], which produces the corresponding periodic broadcast schedule.

3 Simulation Results

In this section we compare the proposed, DCA-HSR scheme with the related approaches of [5] and [15]. The *Service Scheduler* of [5] constitues a classic approach which targets the maximization of the service ratio in a broadcast environment. The *Impatience Scheduler* of [15] on the other hand, follows a contemporary approach of directly minimizing the mean client impatience that a schedule incurs. As already stated, a periodic scheduling process consists of two steps: i) defining the optimal number of occurrences of each class in the schedule, and ii) serializing the data according to the defined occurrences. Each compared study follows its own way of defining the optimal class occurrences. However, all studies follow the generic serialization process of [11] for fairness reasons.

The simulation scenarios consider a varying number of data classes (N), varying class p.d.f. (p_i) skewness and varying data criticality (s). Since the ratio p_i/c_i is already varied through p_i, the class cardinality (c_i) is considered to be equal to one size unit (e.g. *GBytes*), for all i. In each case, the topology consists of a central, broadcast scheduling server and a number of 10^3 tuned-in clients. Firstly, the server produces the optimal number of data class occurrences, v_i^{opt}, according to the employed algorithm. The calculated v_i^{opt} values are fed as input to the serialization process of [11], which has been designed to produce periodic broadcast schedules. The resulting schedule is broadcasted to the clients, fulfilling their demands. The simulation stops when 300 queries per client have been answered or dropped. The waiting time before a client drops a given query is picked randomly from the exponential distribution of (1). After each answer, a client waits for a random *think time* $\in [0, N]$ before posting a new query. The client queries as a whole follow the predefined class p.d.f., p_i. Each simulation is repeated 10^3 times in order to extract a dependable mean value. Concerning the class request probabilities, we employ the ZIPF [13] p.d.f., as in the majority of the cited works. A parameter $\theta \in [0, \infty)$ regulates the skewness of the distribution. The value $\theta = 0$ corresponds to a flat distribution, while higher values indicate an increasingly skewed p.d.f..

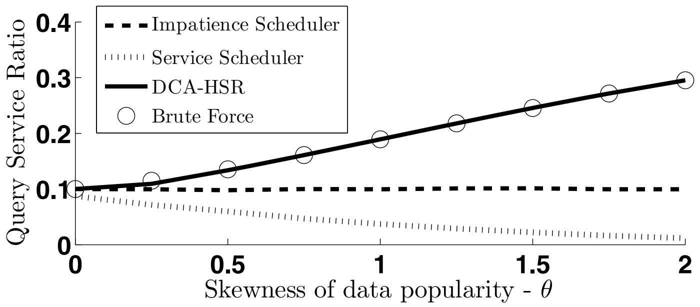


Fig. 2. The behavior of the compared approaches when the commonality in the clients' preferences increases. The proposed DCA-HSR approach focuses on the increasingly popular data classes and achieves near-optimal service ratio in all cases. The Impatience and Service scheduler strive to serve all classes, causing the abandonment of a considerable amount of queries.

Figures 2 and 3 study the case of $N = 5$ available classes. The number is purposefully small, in order to enable comparison with brute force-derived results. Brute forcing constitutes of running the simulation for all possible subsets that can be formed from the original N classes, and keeping the one that achieves the highest service ratio. Notice that there exist 2^N possible subsets in any case. As both figures 2 and 3 illustrate, the proposed DCA-HSR algorithms achieves near-optimal results in all test cases. In detail, Fig. 2 illustrates the performance of the compared approaches when the skewness of the p_i distribution is varied through the θ parameter of the ZIPF formula. The data criticality is kept constant at $s = 2$. Increased θ values correspond to a higher degree of commonality in the clients' demands (i.e. requesting common classes). The proposed DCA-HSR focuses on the common client needs, discarding non-profitable data classes. Thus, it achieves a much higher service ratio than the compared solutions, which strive to disseminate all data classes. When data criticality increases (Fig. 3, $\theta = 1.0$) all solutions suffer from decreased performance, since the client deadlines be-

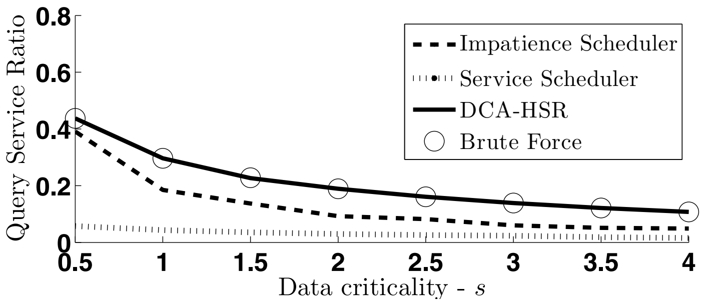


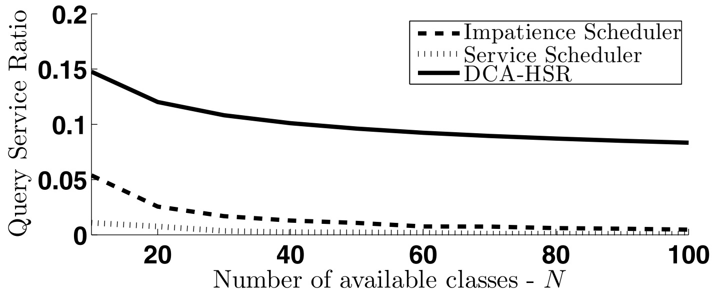
Fig. 3. All compared approaches exhibit lower performance when the data criticality is increased. However, the proposed DCA-HSR approach minimizes the losses by focusing on the most critical data classes.

come more restrictive. However, the proposed DCA-HSR can limit the losses in a nearly-optimal manner, by once again focusing on the most profitable classes.

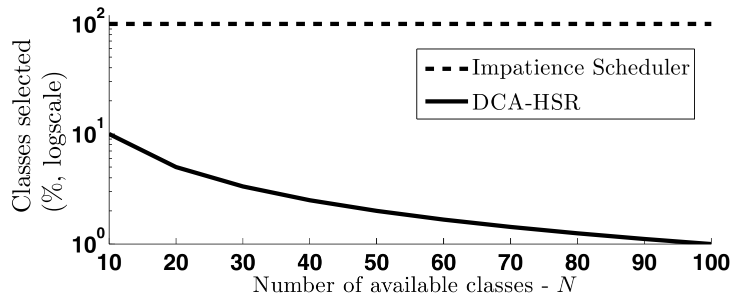
Content selectivity is more evident in Fig. 4 which studies the case of varying the number of available data classes N. The remaining variables are set as $\theta = 1.0$ and $s = 2.0$. Fig. 4a shows the achieved service ratio for each of the compared approaches. Notice that brute forcing is no longer applicable for practical reasons (2^N becomes excessively high). The proposed DCA-HSR algorithm achieves the highest service ratio in all cases $N \in [10, 100]$. What is equally important is the fact that this performance is achieved while using only a small fraction of the original data set (Fig. 4b). Notice that the higher the cardinality of the data set, the higher the aggregate broadcasting cost. Data items must typically be selected and preprocessed by humans in order to achieve sufficient publication quality. This factor aside, the scheduling process itself requires more computational power to serialize the broadcast of the items [11]. Therefore, the proposed DCA-HSR can not only attract more clients, but can actually do so with a much reduced cost.

4 Conclusion and Future Work

The present study proposed the Data Clustering Algorithm for Higher Service Ratio (DCA-HSR) which can choose the subset of the original data that yields near-optimal service ratio. Comparison with related studies showed that the DCA-HSR can attract several times more clients, while employing a small fraction of the available data. This fact can also be used for cutting down on the generic expenses associated with the broadcast process. Future work is directed towards broadcasting the secondary data sets produced by the DCA-HSR in a multichannel scheme, increasing the service ratio further.



(a) Service ratios achieved by the compared approaches. By focusing on promising data subsets only, the proposed DCA-HSR algorithm achieves three to ten times greater performance than the related schemes.



(b) Ratio of selected classes for broadcasting, corresponding to Fig 4a. The Impatience Scheduler and the Service Scheduler attempt to disseminate all available data classes in any case. On the other hand, the proposed DCA-HSR algorithm selects a nearly-optimal subset of the classes for broadcasting.

Fig. 4. The proposed DCA-HSR not only achieves much greater service ratio, but employs a minimal subset of the available data as well. Thus, DCA-HSR achieves both increased performance and minimal scheduling cost (pre-processing data for publication, data serialization [11]).

References

1. Acharya, S., Alonso, R., Franklin, M., Zdonik, S.: Broadcast disks. ACM SIGMOD Record 24(2), 199–210 (1995)
2. Balli, U., Wu, H., Ravindran, B., Anderson, J., Jensen, E.: Utility Accrual Real-Time Scheduling under Variable Cost Functions. IEEE Transactions on Computers 56(3), 385–401 (2007)
3. Chen, J., Lee, V.C.S., Zhan, C.: Efficient Processing of Real-Time Multi-item Requests with Network Coding in On-demand Broadcast Environments. In: Proceedings of the 15th Int. Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA 2009), Beijing, China (August 2009)
4. Gecsei, J.: The Architecture of Videotex Systems. Prentice-Hall, Englewood Cliffs (1983)
5. Jiang, S., Vaidya, N.H.: Scheduling data broadcast to "impatient" users. In: Proceedings of the 1st ACM International Workshop on Data Engineering for Wireless and Mobile Access (MoBiDe 1999), Seattle, Washington, United States, pp. 52–59. ACM (1999)
6. Jiang, S., Vaidya, N.H.: Response time in data broadcast systems: mean, variance and tradeoff. Mobile Networks and Applications 7(1), 37–47 (2002)
7. Xu, J., Tang, X., Lee, W.-C.: Time-critical on-demand data broadcast: algorithms, analysis, and performance evaluation. IEEE Transactions on Parallel and Distributed Systems 17(1), 3–14 (2006)
8. Kakali, V.L., Sarigiannidis, P.G., Papadimitriou, G.I., Pomportsis, A.S.: A Novel Adaptive Framework for Wireless Push Systems Based on Distributed Learning Automata. Wireless Personal Communications 57(4), 591–606 (2011)
9. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 881–892 (2002)
10. Liaskos, C., Petridou, S., Papadimitriou, G.: Cost-Aware Wireless Data Broadcasting. IEEE Transactions on Broadcasting 56(1), 66–76 (2010)
11. Liaskos, C., Petridou, S., Papadimitriou, G.: Towards Realizable, Low-Cost Broadcast Systems for Dynamic Environments. IEEE/ACM Transactions on Networking 19(2), 383–392 (2011)
12. Liaskos, C., Petridou, S., Papadimitriou, G., Nicopolitidis, P., Pomportsis, A.: On the Analytical Performance Optimization of Wireless Data Broadcasting. IEEE Transactions on Vehicular Technology 59, 884–895 (2010)
13. Pietronero, L., Tosatti, E., Tosatti, V., Vespignani, A.: Explaining the uneven distribution of numbers in nature: The laws of Benford and Zipf. Physica A: Statistical Mechanics and its Applications 293(1-2), 297–304 (2001)
14. Vincent Poor, H., Hadjiliadis, O.: Quickest detection. Cambridge University Press, Cambridge (2009)
15. Raissi-Dehkordi, M., Baras, J.S.: Broadcast Scheduling for Time-Constrained Information Delivery. In: Proceedings of the 2007 IEEE Global Telecommunications Conference (GLOBECOM 2007), Washington, DC, USA, pp. 5298–5303 (November 2007)
16. Su, C.-J., Tassiulas, L., Tsotras, V.J.: Broadcast scheduling for information distribution. Wireless Networks Journal 5(2), 137–147 (1999)
17. Vaidya, N.H., Hameed, S.: Scheduling data broadcast in asymmetric communication environments. Wireless Networks 5(3), 171–182 (1999)