This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

On Data Compatibility and Broadcast Stream Formation

Christos K. Liaskos, *Member, IEEE,* Angeliki N. Tsioliaridou, Georgios I. Papadimitriou, *Senior Member, IEEE*, Petros Nicopolitidis, *Senior Member, IEEE* and Andreas S. Pomportsis

Abstract—The Web offers abundant amounts of information anytime, anyplace. Multimedia provision companies bloom and die, seeking to attract a viable share of the data dissemination market. More often than not, failure is attributed to unsuitable content orientation. The present work contributes a novel content specialization scheme for periodic, push-based data streaming services. Its goal is to maximize the ratio of user requests served within their respective deadlines. In this sense, optimality is proven to be achievable only by discarding all requests for certain data items, focusing solely on the streaming of the remaining ones. Thus, the service achieves the content specialization that yields the most clients. The methodology workflow is as follows. By means of mathematical analysis we quantify data compatibility in terms of boosting the system's service ratio. A data selection algorithm is then proposed, which follows a sort-and-segment approach. Through simulations, the algorithm is shown to surpass related solutions, while coinciding with bruteforce results and analytical expectations.

Index Terms—Push, Data selection, Service ratio optimization

1 Introduction

As the Internet expands incorporating an 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, saving network resources by employing broadcasting. Web and classic TV services are typical examples of systems that exploit this commonality per audience, instead of serving shows per demand of each individual client. 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 [1]. Users experience too long waiting times and flee the system. The broadcast service must then specialize its content, excluding certain data items from its program altogether. Returning to the TV example, a channel manager must select from a given pool the shows that will satisfy the preferences and deadlines of most viewers, ensuring the prosperity of the service. The present study addresses this issue in the context of any push-based broadcasting service, presenting a content specialization scheme that maximizes the client service ratio.

Push-based broadcast systems in general [2] seek to disseminate a set of data items to a set of clients. Their uniqueness lies in disregarding individual client queries and operating on statistics per data item only. TV channels, for example, are interested in the popularity of each show and not in explicit, time-dependent client queries. The monitoring of the data item statistics is handled by a separate process, examined as a separate field of study [3]. Once known, the item statistics are passed to a scheduling process which defines the final series of item broadcasts. The exact form of the series depends on the optimization criterion [4]. For example, it has been proven that a periodic schedule minimizes the mean client waiting time [5]. This principle has been adopted as a standard practice by all TV channels, repeating the same kind of shows periodically on a daily basis.

Classic and modern studies on push scheduling aimed at such improvements in service times, but not necessarily at maximizing the number of subscribed users [5], [6], [7]. The issue stems from the fact that no data selection mechanism is offered. Even when presented with a bulk of nonmatching data, the related approaches will attempt to disseminate all of them in a way that optimizes the given criterion. However, the very fact that the total volume of data for broadcast has increased may hinder the upholding of any client deadlines.

The introduction of data selection prior to scheduling is the contribution of the present work. Differentiating from the related studies, the present paper pursues to directly maximize the ratio of successfully answer client queries. The study will show analytically that in order to achieve the maximum client query service ratio, certain data items must be ignored when building the schedule. A novel algorithm extracts the optimal subset of items that yields optimality. The analytical findings are enforced by simulations and brute force results. Furthermore, the proposed scheme is shown to

The authors are with the Department of Informatics, Aristotle University of Thessaloniki, Greece 54124, e-mail: [cliaskos, atsiolia, gp, petros, apombo]@csd.auth.gr.

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

Digital Object Indentifier 10.1109/TC.2013.104 0018-9340/13/$31.00 © 2013 IEEE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

be dominant over data selection based on heuristics. Thus, it is shown that by using the proposed scheme, a push service such as TV broadcasting can attract more clients while managing much less content.

2 Related Work

Broadcasting, in the sense of data advertisement, is an integral function of modern IT. Push-based broadcasting, which is studied in this paper, is considered as the first stage of the data advertising process. Being initially unaware of the existence of new data, the clients do not post queries to the broadcasting server. In other words, the client queries are *silent*. A client simply listens to the broadcast channels, until successful retrieval of a needed item. Thus, the server can only infer request probabilities of the data items, typically in an adaptive manner [3]. Once the clients become familiar with the advertised content, operations can be handed over to a pull-based service [8].

Having acquired this statistical information per datum, the channel proceeds to schedule the data item broadcasts correspondingly. Early studies on push systems showed that periodicity is a prerequisite for optimality [9]. The time interval between two consecutive item occurrences in the schedule must be constant. Authors in [10] defined the optimal item broadcast frequencies that minimize the mean client waiting time. However, this solution required schedules of infinite size, which led to excessive scheduling complexity. Authors in [11] presented a new scheduler, able to operate optimally with schedules of any user-defined size. Apart from defining the optimal item occurrences, a scheduler must include a serializer that achieves periodicity. Optimal serialization has been proven to be NP-Hard and service preemption has been shown to be beneficial under circumstances in [12]. [1] presented a low-complexity serializer that achieved nearly-optimal results.

Apart from analytical approaches to optimal scheduling, heuristic schemes have been introduced as well. The Broadcast Disks model assumes a virtual system of disks rotating around a common axis [2]. The disks represent items with similar request probabilities. A set of imaginary heads reads and serializes data from the disks, thus forming the schedule. Studies have also addressed the issues of multichannel broadcasting [13], serving correlated items [14], pull-push hybrid data dissemination [15] and data indexing for saving power at the client devices [16].

The related studies seek to improve the user experience by optimizing technical aspects, such as the mean query service time or jitter. In contrast, the present work sets the target of maximizing the client query service ratio at all costs. To this end, it is argued that there is little point in broadcasting all the available data if it means less clients. Analysis shows that a dataset must uphold a certain criterion in order to yield maximum service ratio. A novel algorithm is presented, which can detect the subset of the original data items that can yield the maximum service ratio when broadcasted. Comparison with related approaches, simple heuristics, analytic expectations and brute force results demonstrates the optimality of the algorithm. Conclusively, it is shown that broadcasting less data can increase the number of satisfied clients.

3 System Modeling

The study assumes a star topology comprising a central scheduling and broadcasting server and a number of clients in range. For descriptional simplicity we will assume wireless connectivity. However, the analysis covers any similar push system in general. A set of generic, multimedia data items are to be disseminated to the clients. The operation of the system is push-oriented. The clients do not post queries to a server, but rather listen to the broadcast stream, waiting for updates on items that are of interest to them (*hidden query*). However, their attention span is limited; as their waiting time increases, the probability of abandoning the system for another, better service increases as well. Critical data enforce more strict waiting time requirements, while trivial information is associated with more lax restrictions. These parameters are modeled in the form of: 1) The item index, $i = 1 \dots N$. 2) The popularity, p_i, expressing the probability that a given hidden query is associated with item i. 3) The size c_i of item i, measured in bytes. 4) The severity, s_i, of item i.

Related studies [17] model the clients' attention span as an exponential probability distribution,

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

which expresses the probability of a client waiting for time w before abandoning a query. The item-specific attribute s_i regulates the steepness of the distribution. High values correspond to more time-sensitive items.

The per-item modeling is mandatory for push systems, since the communication is strictly one-way [18]. The item popularity distribution p_i and severity metrics s_i can be inferred from user subscriptions [10], Facebook profile data, simple polling or learning automata-based schemes [3]. The present study refers to the scheduling process that follows the estimation of these attributes.

Once the final, periodic schedule has been constructed, it is broadcasted repeatedly over the wireless clients in range. The final schedule has a total size of:

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

where u_i is the integer number of occurrences of item i in the schedule. The interval between two appearances of item i is:

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

As in [19], the item occurrences are also expressed in ratio form, $v_i = {}^{u_i}/_{C}$. As such, the v_i values are real and infinitesimal calculus can operate on them. In order to calculate the integer number of item presences in the final schedule, u_i, one may multiply the ratios v_i with a required schedule size C and apply rounding (i.e. $u_i = [[C \cdot v_i]]$). Typically, large C values are used, making the precision loss due to the rounding trivial [19], i.e. $u_i = [[C \cdot v_i]] \approx C \cdot v_i$ for large C.

4 Analysis

Differentiating from the majority of the related work, the present study will pursue to maximize the total number of the clients subscribed to the system. The task will be accomplished by maximizing the ratio of successfully served queries. Initially, the optimal ratio of item occurrences, v_i, will be defined. Subsequently, it will be proven that optimality is not always possible. We will then establish a procedure that can select the optimal data subset (out of the $2^N - 1$ alternatives) that yields the maximum number of system clients. Thus, the procedure also sets the upper bound of the possible service ratio for a given data set.

4.1 Maximizing the client query service ratio

Let Q be the total number of client queries during the operation of the broadcast system. Given the definition of the item popularity, p_i, a number of $n_i = Q \cdot p_i$ queries will refer to the i^{th} item. However, a client's attention span is given by the probability distribution of equation (1). Therefore, it is possible that a good number of clients will abandon their requests due to item-specific limitations of their attention span. Reminding that the schedule in question is periodic, assume that a client query is set at time $w \in [0, w_i^{max}]$ before the next scheduled broadcast of the requested item. The client must be willing to wait for an interval of at least w in duration, in order to successfully receive the item. Employing the notation $P(.)$ for expressing probability of the event $(.)$:

$$P(query_i\, answered) = \int_{w=0}^{w_i^{\max}} P(arrive\, at\, time\, w) \cdot P(attention \geq w) \cdot dw \tag{4}$$

The probability $P(arrive\, at\, time\, w)$ is uniform for all w, and equals ${}^1/_{w_i^{max}}$. Additionally, the probability $P(attention \geq w)$ is given by integrating the probability distribution of (1) over the interval $[w, \infty)$. Consequently, equation (4) leads to:

$$P(query_i\, answered) = \frac{1 - e^{-s_i \cdot w_i^{max}}}{w_i^{max} \cdot s_i} \tag{5}$$

Noting that $w_{max} = {}^1/_{v_i}$, equation (5) can also be expressed in terms of item occurrence ratio, v_i, inside the schedule:

$$P(query_i\, answered) = v_i \frac{1 - e^{-\frac{s_i}{v_i}}}{s_i} \tag{6}$$

Equation (6) represents the portion of the queries for a given item i, which are successfully answered. Out of the total $n_i = Q \cdot p_i$ queries for this item, only $n_i \cdot P(query_i\, answered)$ will not be abandoned by the clients. The total ratio of answered queries during the operation of the system equals:

$$\mathcal{M} = \sum_{i=1}^{N} p_i \cdot P(query_i\, answered) \tag{7}$$

where $p_i = {}^{n_i}/_{Q}$. Maximizing the service ratio, $\mathcal{M}$, can effectively increase the total number of subscribed clients, since their abandonment rate will be minimized. The objective is to define the optimal item occurrence ratio v_i which maximizes $\mathcal{M}$. Being a case of constrained optimization, the problem can be solved using the method of Lagrange parameters.

Theorem 1. *The optimal item occurrence ratio, v_i^{opt}, that maximizes the client query service ratio, $\mathcal{M}$, follows the relation:*

$$v_i^{opt} = \frac{-s_i}{1 + W\left(\frac{-p_i + V \cdot c_i \cdot s_i}{e \cdot p_i}\right)} \tag{8}$$

where $W(.)$ is the Lambert-W function [20] and V is a constant which is calculated numerically from the expression:

$$\sum_{i=1}^{N} v_i^{opt} \cdot c_i = 1 \tag{9}$$

Proof: Equation (8) comes as the result of the optimization of equation (7) under restriction (2). First of all, notice that (2) can be written in the form of (9) by dividing both parts with C. We then solve (2) for any arbitrary variable v_j, $j \neq i$, and insert the outcome in equation (7). The problem is then transformed to a case of multivariate optimization. The sufficient conditions for the presence of local maxima are:

$$\begin{cases} \frac{\partial \mathcal{M}}{\partial v_i} = 0 \\ H(\mathcal{M}^{(i,j)})\, is\, negative - definite \end{cases} \tag{10}$$

where $H(\mathcal{M}^{(i,j)})$ is the Hessian matrix [21] with elements $\mathcal{M}^{(i,j)} = \frac{\partial^2 \mathcal{M}}{\partial v_i \partial v_j}$. For any $j = 1 \ldots N. j \neq i$, equation (2) yields:

$$\frac{\partial v_i}{\partial v_j} = -\frac{c_j}{c_i} \tag{11}$$

while the condition $\frac{\partial \mathcal{M}}{\partial v_i} = 0$ of (10) yields:

$$\frac{\partial \mathcal{M}}{\partial v_i} = p_i \cdot \frac{\partial \alpha_i}{\partial v_i} + p_j \cdot \frac{\partial \alpha_j}{\partial v_j} \cdot \frac{\partial v_j}{\partial v_i} = 0 \overset{(11)}{\Longleftrightarrow}$$

$$\frac{p_i}{c_i} \cdot \frac{\partial \alpha_i}{\partial v_i} = \frac{p_j}{c_j} \cdot \frac{\partial \alpha_j}{\partial v_j},\ \forall i, j \tag{12}$$

where $\alpha_i = P(query_i\, answered)$ for brevity. Since (12) must hold for every i, j combination, it is derived that:

$$\frac{p_i}{c_i} \cdot \frac{\partial \alpha_i}{\partial v_i} = V \tag{13}$$

where V[1] is constant for all i. Notice that there exists only one case of nullification for $\partial\mathcal{M}/\partial v_i$. Therefore, a local optimum is global as well. Furthermore, regarding the Hessian matrix, it holds that:

$$\frac{\partial^2 \mathcal{M}}{\partial v_i \partial v_j} = 0,\ i \neq j,\ \frac{\partial^2 \mathcal{M}}{\partial v_i^2} = -\frac{s_i e^{-\frac{s_i}{v_i}}}{v_i^3} < 0\ \forall i \tag{14}$$

In other words, the Hessian matrix is diagonal with negative elements. Therefore its eigenvalues are all negative as well and the matrix is by definition negative-definite.

From (6) and (13):

$$1 - \left(1 + s_i \cdot \frac{1}{v_i^{opt}}\right) \cdot e^{-s_i \cdot \frac{1}{v_i^{opt}}} = \frac{c_i \cdot s_i \cdot V}{p_i} \tag{15}$$

We proceed to set:

$$w = -s_i/v_i^{opt} - 1 \tag{16}$$

Equation (15) is transformed as:

$$1 + w \cdot e^{w+1} = \frac{c_i \cdot s_i \cdot V}{p_i} \Longleftrightarrow w \cdot e^w = \frac{c_i \cdot s_i \cdot V}{p_i \cdot e} - \frac{1}{e} \tag{17}$$

From the definition of the Lambert-W function (See Section 4.2 and Fig. 1) we derive:

$$w = W\left(\frac{c_i \cdot s_i \cdot V}{p_i \cdot e} - \frac{1}{e}\right) \tag{18}$$

Through equation (16), (18) leads to the conclusion of Theorem 1. □

4.2 Solvability

While Theorem 1 provides a seemingly optimal solution to the discussed problem, actual solvability is not ensured. The v_i parameters express ratio of item occurrences in a schedule, and must therefore take positive values. Thus, it must hold that:

$$v_i^{opt} \geq 0 \Longleftrightarrow W\left(\frac{-p_i + V \cdot c_i \cdot s_i}{e \cdot p_i}\right) \leq -1,\ i = 1 \ldots N \tag{19}$$

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.

1. The reader may notice that V corresponds to the Lagrange parameter of the homonymous method. A typical formulation would define the quantity $F = \sum_{i=1}^{N} p_i \cdot \alpha_i + V \cdot \left(1 - \sum_{i=1}^{N} v_i \cdot c_i\right)$ and request that $\frac{\partial F}{\partial v_i} = 0$, $\forall i$, directly producing $\frac{p_i}{c_i} \cdot \frac{\partial \alpha_i}{\partial v_i} = V$ of (13). However, this method does not define a *sufficient* optimization condition but rather a *necessary* one. The Hessian matrix approach makes up for this deficiency.

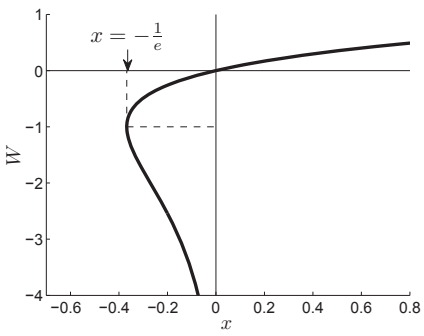


Figure 1: A graphical illustration of the Lambert-W function, $W(x)$. The restriction $W \leq -1$ corresponds to $-1/e \leq x \leq 0$.

The restriction $W(x) \leq -1$ of equation (19) corresponds to $-1/e \leq x \leq 0$. Therefore, for all $i = 1 \ldots N$ it must hold that:

$$-1/e \leq \frac{-p_i + V \cdot c_i \cdot s_i}{e \cdot p_i} \leq 0 \Longleftrightarrow 0 \leq V \leq \frac{p_i}{c_i \cdot s_i},\ \forall i \tag{20}$$

and equivalently:

$$0 \leq V \leq min\left\{\frac{p_i}{c_i \cdot s_i}\right\} \tag{21}$$

Remark 2. Recall from Theorem 1 that V is a constant which is calculated numerically through equations (8) and (9). In this context, relation (21) states that the value set of V is limited by the item with the smallest $\frac{p_i}{c_i \cdot s_i}$ ratio. In other words, if the optimization of Theorem 1 is unsolvable, then the item with minimal $\frac{p_i}{c_i \cdot s_i}$ ratio is to blame. In this sense, this item is incompatible with the others in terms of content.

The second condition of solvability stems from relation (9) of Theorem 1. The restriction can be rewritten in the form of the following equation:

$$f(V_o) = \sum_{i=1}^{N} v_i^{opt}(V_o) \cdot c_i - 1 = 0 \tag{22}$$

where $V_o \in [0, min\left\{\frac{p_i}{c_i \cdot s_i}\right\}]$, in accordance with (21). Notice from Fig. 1 that the Lambert-W function is monotonous for the studied case of $W(V) < -1$. Therefore, the function $v_i^{opt}(V) = \frac{-s_i}{1+W(V)}$ is also monotonous, and so is the summation over all i. Thus, the function $f(V)$ of (22) is monotonous. Consequently, only a single value, $\{V_o | f(V_o) = 0\}$, may exist. Furthermore, the Bolzanno theorem[2] must hold in the range $[0, min\left\{\frac{p_i}{c_i \cdot s_i}\right\}]$. According to it, the function must undergo a sign change:

$$f(0) \cdot f(min\left\{\frac{p_i}{c_i \cdot s_i}\right\}) \leq 0 \tag{23}$$

2. The Bolzanno theorem states that if a continuous, monotonous function $f(x)$ undergoes a sign change in $[a, b]$, i.e. $f(a) \cdot f(b) \leq 0$, then it has exactly one root in this interval and vice-versa.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Theorem 3. *A set of items with attributes* $\{s_i,\, c_i,\, p_i\}$, $i = 1 \ldots N$ *is solvable in terms of maximizing the ratio of served clients,* $\mathcal{M}$*, if it holds that:*

$$\sum_{i=1}^{N} \frac{s_i \cdot c_i}{1 + W\left[\frac{1}{e}\left(\frac{min\left\{\frac{p_i}{c_i \cdot s_i}\right\}}{\frac{p_i}{c_i \cdot s_i}} - 1\right)\right]} > -1 \tag{24}$$

Proof: It must hold that $f(0) \cdot f(min\left\{\frac{p_i}{c_i \cdot s_i}\right\}) \leq 0$ where $f(V) = \sum_{i=1}^{N} v_i^{opt}(V) \cdot c_i - 1$ and v_i^{opt} is given by Theorem 1. For $V = 0^+$:

$$\lim_{V \to 0^+} v_i^{opt}(V) = \frac{-s_i}{1 + W\left(0^+ - 1/e\right)} = \frac{-s_i}{1 + (-1)^-} \to +\infty \tag{25}$$

Therefore, $f(0^+)$ is positive and it should hold that:

$$f(min\left\{\frac{p_i}{c_i \cdot s_i}\right\}) \leq 0 \tag{26}$$

Once expanded, equation (26) yields directly the conclusion of Theorem 3. □

Through relation (24), Theorem 3 provides a way of quickly checking the solvability of a data item set. Furthermore, Remark 2 can be used for pinpointing incompatible items. Thus, an algorithm that detects solvable, optimal subsets of the original items can be formulated.

Algorithm 1 Search for Compatible, Optimal Data Algorithm (SECOD).

INPUT: A set of N items, $\{p_i, c_i, s_i\}$, sorted by ascending $p_i/s_i \cdot c_i$ value.
OUTPUT: The subset yielding the maximum service ratio, **best_subset.**

1: Init $I_1 = 1$; $I_2 = N$;
2: Init $best_subset = []$;
3: Init $best_rate = 0$;
4: **WHILE** $I_2 \geq I_1$
5: Subset $\mathbb{I}^* = I_1 : 1 : I_2$, inclusive;
6: Set $p_i^* = p_i(\mathbb{I}^*)/\sum p_i(\mathbb{I}^*)$; $c_i^* = c_i(\mathbb{I}^*)$; $s_i^* = s_i(\mathbb{I}^*)$;
7: **IF Set_is_solvable**$(\{p_i^*, c_i^*, s_i^*\})$ **THEN**
8: Calculate v_i^{opt} from Theorem 1;
9: Calculate $P(query_i\, answered)$ from equation (6);
10: Calculate $\mathcal{M}$ from equation (7);
11: **IF** $\mathcal{M} > best_rate$ **THEN**
12: Set $best_rate = \mathcal{M}$;
13: Set $best_subset = \mathbb{I}^*$;
14: $I_2 = I_1 - 1$; $I_1 = 1$;
15: **END_IF**
16: **ELSE**
17: $I_1 = I_1 + 1$;
18: **END_IF**
19: **END_WHILE**

4.3 Data subsets that maximize the number of clients

Consider the procedure formulated as Algorithm 1. The procedure receives a set of data items and returns the subset that yields the maximum query service ratio. The algorithm requests that the items be initially indexed by ascending $p_i/s_i \cdot c_i$ value. This requirement is in line with Remark 2: should the given data set be incompatible, the algorithm should discard the item with the smallest $p_i/s_i \cdot c_i$ value first. This sorting also simplifies the definition of the optimal subset as a sequential index range, $[I_1, I_2]$. The algorithm is initialized for checking the solvability of the complete set first ($I_1 = 1$, $I_2 = N$, step #1). The check is performed via Theorem 3 and occurs at step #7 through the pseudo-function *Set_is_solvable()*. If the set is not solvable, the item with the smallest $p_i/s_i \cdot c_i$ value is discarded at step #17 (unary increment of I_1). The subset $\mathbb{I}^*$ is formed according to the resulting indexes at step #5, and the new item attributes are set at step #6. Normalization is also applied to ensure that the sum of the new probabilities equals one. Subset solvability is checked again, and the process is repeated iteratively until a valid subset has been found. Notice that convergence is guaranteed, since a single item is always compatible to itself.

Once a valid subset has been found, the SECOD algorithm proceeds to estimate the achievable query service ratio, $\mathcal{M}$ at steps #8-10. Notice that the calculations of step #10, pertaining to equation (7), include the original probabilities, p_i, and not the normalized ones, p_i^*. The achieved $\mathcal{M}$ value is logged and, should it be better than the previously logged value (step #11), it is registered as the $best_rate$. The corresponding data subset is also registered as the $best_subset$ variable. Both variables were properly initialized at steps #2,3. The algorithm then proceeds to check the subset formed by the discarded items. This subset is defined as $I_2^{'} = I_1 - 1$ (i.e. the last discarded item) and $I_1^{'} = 1$ (the first discarded item) at step #14. The procedure is repeated, unless $I_2^{'} < I_1^{'}$ (step #4) which denotes that the data items have been depleted.

The complexity of SECOD is $O(1/2 \cdot N \cdot (N+1))$.

5 Simulations

In this Section, the SECOD algorithm is evaluated through simulation in the MATLAB environment [22]. For a given set of N data items, SECOD is requested to define the subset that, once broadcasted, yields the maximum query service ratio. The achieved performance is compared to bruteforce results, analytic expectations, related approaches [7], [17] and simple heuristics. Bruteforce consists of simulating all possible $(2^N - 1)$ data subsets and keeping the one that achieves the best service ratio. The analytic expectations of the service ratio are calculated through (7) for the optimal item occurrence ratio of (8).

The SECOD algorithm seeks to optimize the service ratio directly. However, in their totality, related studies seek to increase the service ratio indirectly, by providing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

better quality of service. The studies of [17] and [7] represent this common approach. The *Service Scheduler* of [17] is a classic scheme which targets the maximization of the service ratio by minimizing the mean service time over all queries. The *Impatience Scheduler* of [7] on the other hand, is oriented towards modern psychometrics. The waiting time of a client for a given query, w, is connected to his impatience through an increasing function of the form e^w. The impatience scheduler then seeks to maximize the system's service ratio by minimizing the mean impatience over all queries. The aforementioned approaches will then define the corresponding ratio of item occurrences, v_i, and produce the periodic schedule accordingly. On the other hand, SECOD will initially choose a subset, e.g. $I_1 \subset \{1 \dots N\}$, of the original items and discard the rest. Then, it will set the optimal v_i by equation (8). At the end, it will produce the corresponding periodic schedule.

Finally, due to the absence of studies that support item discarding, we include a simple heuristic in the comparison. Recall from Remark 2 that the $p_i/c_i \cdot s_i$ ratio characterizes the compatibility of the given items. In addition, an item with a very dissimilar $p_i/c_i \cdot s_i$ ratio can break the compatibility of a given dataset. Therefore, a straightforward approach would be to select and broadcast the items with the highest and most similar $p_i/c_i \cdot s_i$ ratios. We therefore employ the K-means grouping algorithm [23] to cluster the items, using the euclidean distance:

$$d_{ij} = |p_i/c_i \cdot s_i - p_j/c_j \cdot s_j| \tag{27}$$

for the data grouping. Since the K-means requires the number of clusters, NoC, as an input, we execute the clustering for every possible $NoC = 1 \dots N$ and keep the group that yields the highest expected service ratio according to equation (7).

As already stated, a periodic scheduling process consists of two steps: i) defining the optimal occurrence ratio of each item in the schedule, and ii) serializing the data according to the defined occurrences. Each compared study follows its own way of defining the optimal item occurrences. However, all studies follow the generic serialization process of [1] for fairness reasons.

5.1 Setup

The simulation runs are executed on a wireless star topology consisting of a central broadcasting server and 10^3 tuned-in clients. All clients are assumed to be in the server's range. Following a common convention ([1], [2], [3], [5], [6], [7]), wireless propagation issues are idealized in order to focus on the evaluation of the scheduling processes only. Furthermore, the one-way communication paradigm from a central, potent server to the clients limits the effects of such issues, as in TV broadcasting.

The server model relies on push-based broadcasting. Once a schedule of data transmissions has been constructed by the given algorithm, the server begins broadcasting the data accordingly. The broadcast is periodic in the sense of constant item broadcast frequencies, as well as in the sense of unceasing repetitions of the complete schedule [1]. The server does not accept input from the clients and it is assumed that the data attributes, {popularity, size, severity} have been approximated statistically in advance [3].

The original data set consists of a varying number of data items (N), with varying popularity (p_i), size (c_i) and severity (s_i) per run. All values are picked randomly from the $[0, 1]$ interval uniformly, applying normalization where applicable. The item sizes are measured in $GBytes$, while the item popularity and severity attributes are unitless.

A client has a series of queries referring to the available items. For any given query, the client waits for the broadcast of the corresponding item without posting additional ones. Depending on the severity of the requested data, s_i, the client may abandon a query before its completion and proceed to the next query. The waiting time before a client drops a given query is picked randomly from the exponential distribution of (1). In accordance with [2], if the query is answered successfully, a client waits for a uniformly random *think time* $\in [0, C]$ before posting a new query (C is the total schedule size). Notice that the required C value must be provided by the user. As shown in [11], C regulates the efficiency of the produced schedule as a function of the required computational resources for its creation. We set $C = 10 \cdot \sum_{i=1}^{N} c_i$, which has been shown to constitute a good choice from that aspect [1]. During the initialization phase, $3,000,000$ queries are generated according to the the predefined item popularity distribution, p_i, and assigned equally to the 10^3 clients. The employed pseudo-random number generator engine is the Mersenne Twister with random seeds per simulation run. The simulation terminates when 3000 queries per client have been answered or dropped. The typically achieved confidence interval is 99% or better.

In summary, the simulation inputs are the item attributes, $\{p_i, c_i, s_i\}, i = 1 \dots N$, and the output is the query service ratio. The simulation duration corresponds to the "grace" interval given from the clients to the system. In other words, the clients do not abandon the system during the simulation. However, it is assumed that the achieved query service ratio defines the number of clients in a proportional manner. Should an algorithm yield a service ratio of e.g. 30% during this period, it is assumed that it translates to 70% of the clients fleeing the system in the long ran. Specialized client modeling may upset this proportionality. For example, client models assuming multiple concurrent queries that are generated and dropped asynchronously might not exhibit strict proportionality. Nonetheless, the relation connecting the query service ratio and the number of subscribed clients is strictly rising in any case. Thus, through a simple, case-specific mapping, the presented conclusions apply to any specialized case of this kind.

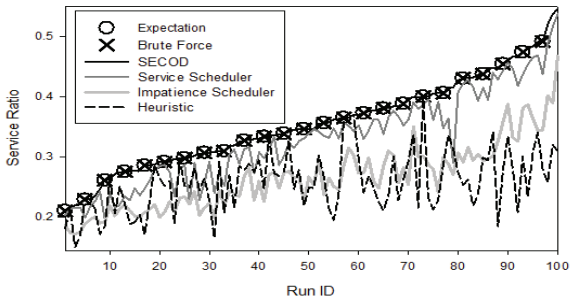


Figure 2: Performance of SECOD for $N = 10$ data items. SECOD coincides with the analytical expectations and the bruteforce results in all cases. The related approaches behave suboptimal in all runs.

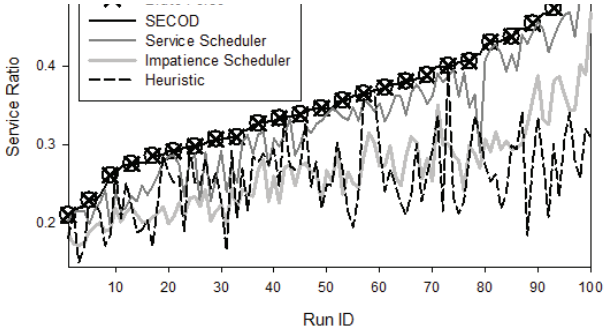
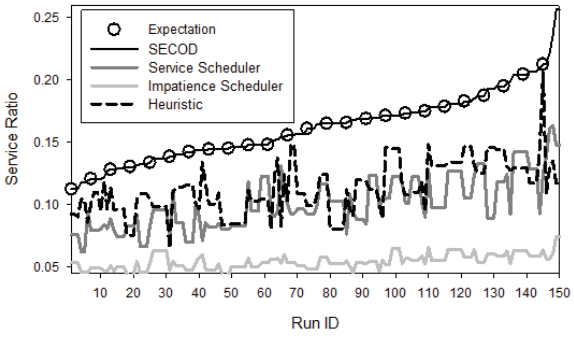


Figure 3: Raising the number of data items to $N = 20$ does not change the relative ranking of the compared schemes. However, the difference between SECOD and the related approaches is magnified.

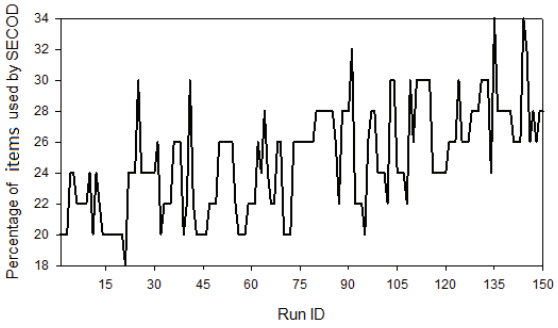
5.2 Results

Figures 2, 3 and 4 illustrate the query service ratio achieved by the compared schemes in the cases of $N = 10$, 20 and 50 data items respectively. Each case comprises several simulation runs for randomized inputs $\{p_i, c_i, s_i\}$. The query service ratios achieved by the competing schemes are logged and sorted by ascending performance of the best scheme. In the case of $N = 10$ data items we are able to derive the optimal performance through bruteforce. For every set of $N = 10$ items there exist $2^N - 1$ possible subsets. Thus, for every run, we conducted $2^{10} - 1$ simulations and kept the subset that performed best. As Fig. 2 illustrates, SECOD was able to converge to the optimal subset in all runs, achieving the highest possible service ratio. The service scheduler produced the second-best performance, while the impatience scheduler came last in all runs. Both of the related schedulers attempt to disseminate the complete data set. However, the service scheduler takes query abandoning into account and focuses on the maximization of the client ratio. On the other hand, the impatience scheduler assumes that the waiting time for a given item can potentially incur more impatience than another. Thus, it can assign degrees of delivery time sensitivity to each item, a feature not supported by the service scheduler. The objective is then to minimize the mean overall impatience. Evidently, targeting the maximization of the query service ratio is more efficient than providing optimal quality of service. SECOD combines both QoS capabilities (through the item-specific severity attribute. s_i) and service ratio-targeted performance. Moreover, it introduces optimal content selection, which is not supported by the related studies.

The number of $N = 10$ data items makes for 1023 possible subsets. However, an algorithm should be able to pinpoint the optimal subset out of larger pools. Figure 3 examines the case of $N = 20$ items, which yields $\sim 10^6$ possible subsets. In this case, bruteforcing was not feasible for computational reasons. Nevertheless, the performance ranking of the compared algorithms



(a) At $N = 50$ the difference between SECOD and the service and heuristic schedulers becomes substantial, approaching 100%. The performance of the impatience scheduler reduces to near-static.



(b) Explaining the difference in performance. SECOD selects and broadcasts only an optimal 18 to 34% of the available data items, thus achieving better performance for less service.

Figure 4: Performance of the compared schemes for $N = 50$ data items with random attributes.

remained unaltered in all runs. SECOD produced the best results in all cases. On the other hand, the relative difference in the performance of SECOD and the service scheduler begins to widen. We proceed to increase the number of available items to $N = 50$ in Fig. 4.

As shown in Fig. 4a the performance of SECOD becomes approximately two times better than the service

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

and heuristic schedulers in all runs. Furthermore, the impatience scheduler does not exhibit much scalability in terms of N, reverting to a nearly-static kind of performance. This fact highlights the need for optimal data selection, especially in large original sets. Notice in Fig. 4b that SECOD uses only a limited portion of the available data and yet it achieves the best service ratio in all runs. The portion of used data does not exhibit a predictable correlation to the service ratio plot of Fig. 4a. Recall that the item attributes are randomized in each run, while SECOD applies an exquisite criterion for the data selection. Therefore, in Fig. 4a optimality may require just as low a portion as 18% of the available data, or as high as 34%. Notably, even the simple, k-means heuristic performs equally or better than the related approaches. This fact highlights the positive effects of proper content selection. In essence, a push system can satisfy more clients when it specializes in carefully selected content.

Finally, it is noted that the presented version of SECOD assumes exponentially distributed item severity (eq. (1)). Nonetheless, the analytical steps remain the same for other models as well. The only prerequisite is the corresponding update of the $P(query_i\, answered)$ probability in equation (5). The extracted conclusions also hold when a client is more than one hops away from the server. The client will then simply receive the same stream of data at an offset of a few $msec$. Stream buffering also falls into this case. Since the interarrival of client requests is random and the users are independent of each other, the end result remains the same. However, the notion of the "in-range" client is altered. Multihop networking issues, such as congestion and extreme jitter may lead to faulty stream reception. A client with unacceptably low reception quality will not be able to rate the stream contents and will eventually be excluded from the statistical monitoring process. Node mobility also affects the service coverage in the same manner. Future work aims at taking into account such effects in the form of additional statistics per item, eventually considering factors such as reception rate, copyright costs and advertising gains alongside the service ratio.

6 Conclusion

The present work introduced a novel way of data selection for optimal service ratio in wireless broadcast systems. The analysis concluded that a set of data items must meet certain low-level conditions in order to be appropriate for optimal broadcast. Furthermore, a novel algorithm was derived which can pinpoint the data subset that yields the maximum query service ratio. Comparison with related approaches, analytical expectations, heuristics and bruteforce results demonstrated the superiority of the proposed scheme.

References

[1] C. Liaskos, S. Petridou, and G. Papadimitriou, "Towards Realizable, Low-Cost Broadcast Systems for Dynamic Environments," *IEEE/ACM Trans. on Networking*, vol. 19, no. 2, pp. 383–392, 2011.

[2] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks," *ACM SIGMOD Record*, vol. 24, no. 2, pp. 199–210, 1995.

[3] V. L. Kakali, P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "A Novel Adaptive Framework for Wireless Push Systems Based on Distributed Learning Automata," *Wireless Personal Communications*, vol. 57, no. 4, pp. 591–606, 2011.

[4] U. Balli, H. Wu, B. Ravindran, J. Anderson, and E. Jensen, "Utility Accrual Real-Time Scheduling under Variable Cost Functions," *IEEE Trans. on Computers*, vol. 56, no. 3, pp. 385–401, 2007.

[5] J. Gecsei, *The Architecture of Videotex Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] J. Chen, V. C. S. Lee, and C. Zhan, "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'09), Beijing, China, Aug*, 2009.

[7] M. Raissi-Dehkordi and J. S. Baras, "Broadcast Scheduling for Time-Constrained Information Delivery," in *Proceedings of the 2007 IEEE Global Telecommunications Conference (GLOBECOM'07), Washington, DC, USA, November*, 2007, pp. 5298–5303.

[8] M. Bhide, P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham, and P. Shenoy, "Adaptive push-pull: disseminating dynamic Web data," *IEEE Trans. on Computers*, vol. 51, no. 6, pp. 652–668, 2002.

[9] M. Ammar and J. Wong, "On the Optimality of Cyclic Transmission in Teletext Systems," *IEEE Trans. on Communications*, vol. 35, no. 1, pp. 68–73, 1987.

[10] N. H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," *Wireless Networks*, vol. 5, no. 3, pp. 171–182, 1999.

[11] C. Liaskos and G. Papadimitriou, "Generalizing the Square Root Rule for Optimal Periodic Scheduling in Push-based Wireless Environments," *IEEE Trans. on Computers*, vol. [preprint] (d.o.i.: 10.1109/TC.2012.30), 2012.

[12] D. Serpanos, "Scheduling objects in broadcast systems with energy-limited clients," *Computer Communications*, vol. 27, no. 10, pp. 1036–1042, 2004.

[13] B. Zheng, X. Wu, X. Jin, and D. L. Lee, "TOSA: a near-optimal scheduling algorithm for multi-channel data broadcast," in *Proceedings of the 6th International Conf. on Mobile Data Management (MDM'05), Ayia Napa, Cyprus, May*, 2005, pp. 29–37.

[14] Chih-Lin Hu and Ming-Syan Chen, "Online Scheduling Sequential Objects with Periodicity for Dynamic Information Dissemination," *IEEE Trans. on Knowledge and Data Engineering*, vol. 21, no. 2, pp. 273–286, 2009.

[15] S. Kim and S. H. Kang, "Scheduling Data Broadcast: An Efficient Cut-Off Point Between Periodic and On-Demand Data," *IEEE Communications Letters*, vol. 14, no. 12, pp. 1176–1178, 2010.

[16] H.-Y. Shin, "Exploiting skewed access and energy-efficient algorithm to improve the performance of wireless data broadcasting," *Computer Networks*, vol. 56, no. 4, pp. 1167–1182, 2012.

[17] S. Jiang and N. H. Vaidya, "Scheduling data broadcast to "impatient" users," in *Proceedings of the 1st ACM International Workshop on Data Engineering for Wireless and Mobile Access (MoBiDe'99), Seattle, Washington, United States*. ACM, 1999, pp. 52–59.

[18] C. K. Liaskos and G. I. Papadimitriou, "Optimal Periodic Scheduling Under Multimodel Per-Item Constraints in Wireless Systems," *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1071–1080, 2012.

[19] S. Hameed and N. H. Vaidya, "Efficient algorithms for scheduling data broadcast," *Wireless Networks*, vol. 5, no. 3, pp. 183–193, 1999.

[20] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the LambertW function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[21] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, "Generalized Hessian matrix and second-order optimality conditions for problems withC 1,1 data," *Applied Mathematics & Optimization*, vol. 11, no. 1, pp. 43–56, 1984.

[22] MathWorks-Inc, "MATLAB version (R2010a)," 2010. [Online]. Available: http://www.mathworks.com/products/matlab/

[23] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.