Optimal Periodic Scheduling Under Multimodel Per-Item Constraints in Wireless Systems

Christos K. Liaskos, *Member, IEEE*, and Georgios I. Papadimitriou, *Senior Member, IEEE*

***Abstract*—A novel periodic scheduling scheme for wireless environments is presented. The proposed scheme enables quality of service (QoS) agreements per data item, contrary to the per-dataset approach of existing solutions. Constraints abiding by any of the existing models (impatience, utility, and waiting time) are allowed to exist concurrently. The impact of individual agreements on the system's performance as a whole is analytically studied and quantified. Comparison with related approaches rendered the proposed scheme optimally efficient and flexible enough to serve as a basis for the implementation of specialized QoS metrics as well.**

***Index Terms*—Periodic push, quality of service (QoS), wireless systems.**

I. Introduction

DATA broadcasting is an efficient technique for preserving bandwidth, especially in wireless environments. The steady increase in the volume of data requested by the end users calls for direct exploitation of query similarities. However, a large volume of data typically includes multiple classes of information. Proper quality of service (QoS) dictates that each data item must be handled individually, according to its own class model, and that several models must be respected concurrently. Furthermore, at least for practical purposes (e.g., pricing), the impact of each individual QoS constraint on the overall system must be defined. To the best of the authors' knowledge, these issues in periodic broadcast scheduling have not been discussed prior to this work.

The discussed push/broadcast technology [1] generally describes a way of communication where the clients do not post explicit queries to the server. The clients are interested in a common set of data items with a known request probability distribution. The probabilities are extracted either directly, by means of a feedback scheme, or indirectly, e.g., by studying Facebook profile data. A central authority then creates a broadcast schedule that matches the needs of the clients optimally in a statistical manner. The push approach is contrasted with pull technology [2]–[4] where the clients send explicit requests to a server. The server then serializes his answers in an optimal manner [5], [6]. Notice that the push approach inherently treats all clients as equals, because of the lack of explicit queries. Therefore, QoS issues and agreements refer to the data items and not to the clients. The QoS per item is defined centrally or agreed upon in an auction-style transaction among the clients.

In terms of system setup, wireless data broadcasting typically considers a large number of clients dispersed over an area covered by a wireless network. The functionality of the system relies on a server publish/client subscribe logic [7]. A central publishing server administers a repeating broadcast schedule of data items to all clients in the covered area. The server periodically edits this program by increasing or decreasing the number of occurrences of each item, or by adding new ones in an effort to keep it in line with the clients' demands. A client simply subscribes to the service and reads data from broadcast stream. The clients' preferences, which are expressed in the form of an item request probability distribution, may be static. However, in the most common scenario, they are periodically estimated by the server through an adaptation scheme [8]. Push-based broadcasting is commonly applied to teletext services, as well as video and audio broadcasting [9], typically in conjunction with a content indexing scheme [10].

As in most contemporary communication systems, providing individualized QoS is becoming a new trend [11]. However, related studies on broadcast systems have addressed system-wide issues only. The optimization of the mean waiting time [12] and the jitter of the waiting time [13] over all clients, as well as the mean impatience for data availability [14] and the global data utility [15], have constituted notable goals. Thus, meeting item-specific constraints is not possible with existing scheduling techniques. For example, keeping the utility of a specific item above a given level is not accounted for. In addition, different QoS models, such as the cited ones referring to impatience, utility and jitter, cannot be mixed in one scheduling process. All QoS agreements have to involve a single common model, excluding all others. This issue creates practical contradictions. As an example on TV scheduling, delaying the broadcast of an unpopular soap opera must be assumed to cause impatience in the same manner as a national disaster. Moreover, the impact of holding an individual QoS agreement has not been studied either, even though it is a prerequisite for practical applications, such as monetary pricing of the service provided to the end user.

This work proposes a solution to the aforementioned issues. The study comprises of the following steps: It is initially proven that a data-item-specific constraint abiding by any model can be substituted by an equivalent restriction involving only the item's maximum serving time. Thus, mixed-model

Manuscript received September 21, 2011; revised December 23, 2011; accepted February 28, 2012. Date of publication May 1, 2012; date of current version October 12, 2012. This work was supported 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—Research Funding Program: Heracleitus II. Investing in knowledge society through the ESF. This paper was recommended by Associate Editor N. Wu.

The authors are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece (e-mail: cliaskos@csd.auth.gr; gp@csd.auth.gr).

Digital Object Identifier 10.1109/TSMCC.2012.2190927

1094-6977/$31.00 © 2012 IEEE

individualized QoS is enabled. Items with no associated constraints are also accounted for. Furthermore, it is shown that the presence of constraints has a direct impact on the global mean waiting time. Formulas connecting these quantities are given. Simulations prove the formulas to be accurate in all test cases. Comparison with prior QoS schemes highlights the flexibility of the proposed scheduling scheme in terms of efficiency in meeting item-specific constraints.

The remainder of this paper is organized as follows. Section II presents an overview of related work on wireless broadcast systems. Section III discusses standard assumptions and theoretical prerequisites. The proposed QoS framework is described in Section IV, and its evaluation takes place in Section V. Conclusion is given in Section VI.

II. Related Work

Research on wireless broadcast systems initially focused on the minimization of the clients' mean waiting time. In [12], it was proven that the mean waiting time depends solely on the popularity (request probability) and size of each data item, and not on the number of clients. A lower bound for the mean waiting time was given, as well as scheduling algorithms that achieved it at the expense of increased complexity. These algorithms worked only for nearly equally sized items. In [16], the authors presented an analysis-derived periodic scheduler that achieved the same performance with minimal complexity for any item sizes. The broadcasting process was divided into a scheduling and a serializing stage. The scheduling stage defines the optimal number of periodic item occurrences in the schedule, which yield minimal waiting time. The serializing process is generic, receiving any item occurrences and producing the corresponding series of actual periodic item broadcasts. The low complexity of [16] pertains to both stages as a whole.

Heuristic low complexity scheduling methods were introduced in [1] with the introduction of the broadcast disks model. According to it, items are grouped by popularity, forming virtual disks rotating around a common axis. Imaginary heads read and serialize data from the disks, producing the final schedule. In [17], the authors applied clustering techniques to perform the data grouping. In [18], the grouping of items was analytically optimized and the results were exploited to produce minimal-complexity schedulers. All these studies focused on the minimization of the clients' mean waiting time.

An initial attempt to address QoS issues in broadcast systems took place in [13] where the authors presented a way of minimizing the variance of the clients' waiting time. Modified versions of the algorithms in [12] were demonstrated, which achieved tunable tradeoff 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 [19], with a particular interest in impatience [2], [14], [20], [21]. The optimization procedure follows the pattern of [12], typically applying the method of Lagrange multipliers, substituting the waiting time formula with a study-specific impatience metric. The impatience for a specific item is generally a rising function of the waiting time. In [22], the authors combined pull–push hybrid schemes, presenting ways of optimally choosing which data items are to be disseminated in each of these fashions. In [14], the authors presented the first related algorithm that differentiated between delivery-time sensitive and nonsensitive data items, as is most common in realistic conditions. However, meeting item-specific constraints is impossible, as in all related studies. Furthermore, this study too enforced the same study-specific deadline model on all items, hindering its use as a generic QoS framework.

Psychometrics have also been presented in the form of utility functions [15], [23]–[25]. The usefulness of a received data item is considered to be a function of the waiting time that is required to become available. Several types of utility functions have been used, ranging from rectangular pulses [24]—which are also known as strict deadline functions—to exponential forms or other similar smooth functions [15], [25]. The latter are also known as soft deadline functions. The impatience and utility metrics are opposites, and studies that utilize them aim at minimizing the former or maximizing the latter in a complementary fashion. It must be stated that both metrics are defined arbitrarily, not relying on medical research or user statistics.

Differentiating from the preceding studies, this study introduces a QoS-aware scheduling scheme that allows the concurrent satisfaction of data-item-specific constraints that obey to any (or no) model. The technical contribution comprises formatting the QoS restrictions flexibly, quantifying their impact on the system as a whole, and upholding them accurately. In particular, we study the ability to express the QoS restrictions in a variety of popular formats, e.g., specific mean value or variation of waiting time, impatience, or utility of an item. We show that meeting some restrictions impedes the formation of new ones. Subsequently, the QoS pricing by the provider must vary accordingly. Therefore, we introduce a reference point of operation and a technical metric that quantifies the projected impact of QoS restrictions on the efficiency of the system. Finally, we define a scheduling process to meet the active restrictions accurately. Once the optimal number of periodic item occurrences is set, the generic serialization process of [16] produces the final broadcast series.

III. Theoretical Prerequisites and Standard Assumptions

We regard a set of N data items arbitrarily indexed by $i = 1, \ldots, N$. Each item i is associated with its size l_i (in bytes) and its request probability p_i. Additionally, $v_i \in \mathbb{N}^*$ will denote the number of periodic occurrences of item i in the employed broadcast schedule. The total size of the schedule is given by

$$L = \sum_{i=1}^{N} v_i \cdot l_i. \tag{1}$$

No assumptions are made concerning the nature of a data item during the analysis. In accordance with the majority of the related work, an item is simply a piece of information that a client may acquire through a single query. It is clarified that in periodic broadcast scheduling, the term *client query* does not imply posting a request to a server, but rather waiting for the broadcast of a specific item. As shown in Fig. 1, in periodic

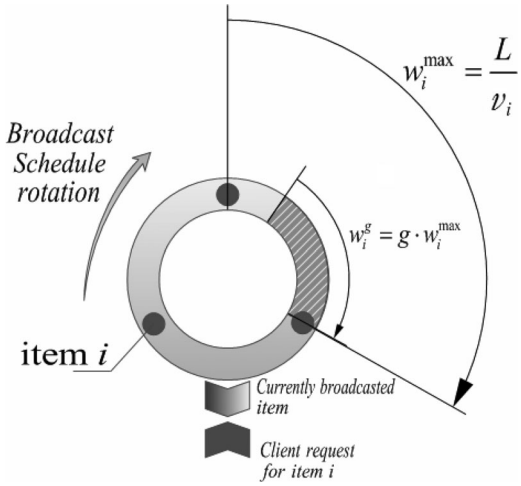


Fig. 1. Periodic schedule containing $v_i = 3$ occurrences of item i. A client request for this item will be answered in time $w_i \in [0, w_i^{\max}]$. $w_i^{\max}$ is the interval between any two consecutive appearances of an item i in a periodic schedule.

schedules, a client query for any item i may be answered in time $w_i \in [0, w_i^{\max}]$ (uniformly distributed, assuming Poisson arrivals and a large number of clients [12], [16]). According to Fig. 1, for the periodic schedules in discussion, it holds that

$$w_i^{\max} = \frac{L}{v_i}. \tag{2}$$

The mean value of w_i with regard to probability distribution p_i is expressed as [12]

$$E_p[w_i] = \sum_{i=1}^{N} p_i \cdot \left(\frac{w_i^{\max}}{2}\right) = \frac{L}{2} \sum_{i=1}^{N} \frac{p_i}{v_i}. \tag{3}$$

Notice that the mean waiting time $E_p[w_i]$ does not depend on the number of clients, which are handled collectively as a Gaussian process via the central limit theorem [12]. In addition, (3) measures the mean waiting time in *size* units (e.g., bytes). Conversion to time units requires knowledge of the physical transmission rate. According to [1], [2], [12]–[18], and [20]–[25]: 1) The assumption of wireless environment is used to justify synchronous data stream reading by all clients. 2) Wireless coverage issues are idealized in order to focus on the evaluation of the scheduling process only. 3) No specific application context is strictly required (e.g., WiFi, Sensor Net, or other). The proposed algorithms can be applied anywhere that scheduling makes sense.

The v_i parameters are chosen to satisfy some predefined optimization criteria. A popular goal is the minimization of $E_p[w_i]$. It has been shown that this criterion is satisfied only through periodic schedules where the v_i parameters satisfy the condition [12]:

$$v_i^o \propto \sqrt{\frac{p_i}{l_i}} \qquad \forall i = 1, \ldots, N. \tag{4}$$

Equation (4) is known as *the square root rule*. The "$(.)^o$" notation denotes optimal values and "$\propto$" proportionality. Since v_i^o must be integers, relation (4) usually holds by approximation.

TABLE I
SUMMARY OF NOTATION

DATA BROADCAST SCHEDULING NOTATION	
Symbol	Explanation
N	The total number of available data items.
$i = 1 \ldots N$	The index of the i^{th} data item (arbitrarily enumerated).
$p_i \in [0, 1]$	The request probability (popularity) of item i: $\sum_{i=1}^{N} p_i = 1$
v_i	The number of occurrences of item i in a periodic broadcast schedule.
l_i	The size of data item i.
L	The size of the whole periodic broadcast schedule: $L = \sum_{i=1}^{N} v_i \cdot l_i$.
w_i	A random variable representing the waiting time for item i. w_i is uniformly distributed in $[0, w_i^{max}]$
w_i^{max}	The maximum waiting time for item i. It is equal to the interval between two consecutive occurrences of item i in the periodic broadcast schedule (constant).
$w_i^{(g)}$	A value of w_i for which $P(w_i \leq w_i^{(g)}) = g$
S_{Total}	Alias for the complete set of available data items.
S_{QoS}	Subset of S_{Total}, containing items with associated QoS constraints.
θ	The skewness factor of any Zipfian p.d.f. ($p_i \propto 1/i^{\theta}$)
STANDARD MATHEMATICAL NOTATION	
$E_*[.]$	The mean value of random variable (.) with regard to probability distribution ($*$).
$P(.)$	The probability of condition (.), $P(.) \in [0, 1]$
$argmax_{(*)}\{.\}$	The maximum value of variable ($*$) for which the expression (.) holds.

By setting $v_i = v_i^o$ in (3), the lower bound of $E_p[w_i]$ is derived:

$$E_p[w_i]_{\min} = 0.5 \cdot \left(\sum_{i=1}^{N} \sqrt{p_i \cdot l_i}\right)^2. \tag{5}$$

The bound of (5) is achievable in practice, as shown in [12] and [16].

The employed notation is summarized in Table I.

IV. Proposed Quality-of-Service-Aware Scheduling Scheme

QoS in push-based systems differs substantially from pull-based ones. In the latter, a server identifies the client and the query type and prioritizes the provided service according to a predefined agreement. This pull-based approach assumes the establishment of a connection and the posting of an explicit query from the client to the server. However, in the studied push-based systems, there exists no connection between the client and the server. The client simply waits until the needed data item is broadcasted by the server, without ever posting any queries. Thus, the server is physically unable to discriminate between the clients. Television broadcasting is a classic example of this category of push systems. Therefore, while QoS in pull-based system refers to clients, in push-based systems, it refers to the broadcasted data. The central scheduling authority typically processes statistical data on the users' preferences, taking into account the cost they are willing to pay for a better QoS. Thus, a common QoS level per data item is derived. The latter can also be predefined via an auction type of transaction among the clients [26]. While the actual definition of such a system of transactions goes beyond our scope, it is notable that game theory could be employed to optimize the profits of the clients or the server [27].

The presentation of the proposed QoS-aware scheduling algorithm (QoSA) follows three steps.

1) Presentation of the procedure that translates any model-specific constraint for an item to a condition involving only its associated waiting time. Conversion to a common format allows for flexibly using any convenient expression for the QoS restrictions.
2) Not all items may have constraints. The next step is to produce a schedule that handles these items in a way that minimizes the clients' mean waiting time, while upholding all existing constraints.
3) Finally, a metric of the impact of the constraints' existence on the global mean waiting time is given. This metric can be used, e.g., to define the cost of upholding a constraint. Any connection with actual monetary cost is, of course, context related and strictly beyond the scope of this paper.

A. Translating Model-Specific Constraints in a Common Format

The most rudimentary constraint format for an item i is the one stating its maximum waiting time (see Fig. 1). This format-denoted as *Format 1*-can be expressed as follows:

Format 1: $w_i^{\max} = W_i$, where $W_i > 0$ is user-defined.

An alternative format may limit the waiting time inside an interval $[0, w_i^{(g)}]$, with a given guarantee $g \in [0, 1]$. This format can be expressed as follows:

Format 2: $P(w_i \leq w_i^{(g)}) = g$, where $P(.)$ denotes probability. It is obvious from Fig 1 that $\frac{w_i^{(g)}}{w_i^{\max}} = g$. Therefore, Format 2 can be translated to Format 1:

$$w_i^{\max} = \frac{w_i^{(g)}}{g}. \tag{6}$$

Another practical constraint format is the definition of the mean waiting time for an item i as

Format 3: $E_u[w_i] = \overline{W_i}$, where u expresses the uniform continuous distribution of $w_i \in [0, w_i^{\max}]$, and $\overline{W_i}$ is a predefined positive value. This format can also be translated to an equivalent expression of Format 1. The mean waiting time for item i is given by

$$E_u[w_i] = \int_0^{w_i^{\max}} w_i \cdot P(w_i) \cdot dw_i. \tag{7}$$

However, since the probability distribution u is uniform over $[0, w_i^{\max}]$ (see [12]), it will hold that $P(w_i) = \frac{1}{w_i^{\max}}, \forall i = 1 \ldots N$. Thus, (7) is transformed as

$$E_u[w_i] = \frac{1}{w_i^{\max}} \cdot \int_0^{w_i^{\max}} w_i \cdot dw_i = \frac{w_i^{\max}}{2} \tag{8}$$

and $w_i^{\max} = 2 \cdot E_u[w_i]$. Therefore, Format 3 is translated to Format 1 as

$$E_u[w_i] = \overline{W_i} \Rightarrow w_i^{\max} = 2 \cdot \overline{W_i}. \tag{9}$$

Equations (6) and (9) are summarized as follows.

Proposition 1: A constraint of waiting time variation (Format 2) or mean value (Format 3) can be translated to a constraint of maximum waiting time (Format 1).

Proposition 1 will be employed to prove that constraints that are based on other metrics (e.g., impatience and utility) have an equivalent "Format 2" or "Format 3," and ultimately "Format 1" expression.

Consider an item-specific metric M_i that is expressed as a function of the waiting time for an item i w_i:

$$M_i = M_i(w_i). \tag{10}$$

Examples are the item-impatience [14] and item-utility [15] metrics. We proceed to make the common assumption [2], [14], [15], [21], [24], [25] that the metric M_i is a monotonous function of w_i.

Proposition 2: Constraints with regard to metric M_i can be translated to Formats 1, 2, or 3 as follows:

1) Constraints of the form $M_i^{\max} = \mathcal{M}_i$ to Format 1:
$$w_i^{\max} = \arg\max_{(w_i)} \{M_i(w_i) = \mathcal{M}_i\} \tag{11}$$
where $\mathcal{M}_i$ is a user-defined value, and $\operatorname{argmax}_{(w_i)}\{M_i(w_i) = \mathcal{M}_i\}$ is the greatest w_i value (w_i^*) for which it holds that $M(w_i^*) = \mathcal{M}_i$.
2) Constraints of the form $P(M_i \leq M_i^{(g)}) = g$ to Format 2:
$$P(w_i \leq \arg\max_{(w_i)} \{M_i(w_i) = M_i^{(g)}\}) = g \tag{12}$$
where $g \in [0, 1]$ and $M_i^{(g)} > 0$ are user-defined.
3) Constraints of the form $E_u[M_i] = \overline{M_i}$ to Format 3:
$$E_u[w_i] = 0.5 \cdot \arg\max_{(w_i^{\max})} \times \left\{ \frac{\int_0^{w_i^{\max}} M_i(w_i) \cdot dw_i}{w_i^{\max}} = \overline{M_i} \right\} \tag{13}$$
where $\overline{M}_i > 0$ is user-defined.

Proof: See Appendix A.

Notice that the expressions $M_i^{\max} = \mathcal{M}_i$ and $P(M_i \leq M_i^{(g)}) = g$ correspond to increasing functions $M_i(w_i)$, such as the impatience metric. If $M_i(w_i)$ is decreasing (e.g., utility metric), the appropriate notation would be $M_i^{\min} = \mathcal{M}_i$ and $P(M_i \geq M_i^{(g)}) = g$. Expressions (11)–(13) remain unaltered in any case. ■

As a consequence of Propositions 1 and 2, any metric-specific constraint can be converted to a Format 1 compatible form (i.e., $w_i^{\max} = W_i$). For the remainder of the analysis, it is assumed that all constraints have been converted to this form. Notice that the scheduler of [16] can then carry out the actual broadcast serialization process, requiring only this input.

B. Meeting the Existing Constraints While Minimizing the Total Mean Waiting Time

Not all N items in the total set S_{Total} may have constraints. Only a subset $S_{\text{QoS}} \subseteq S_{\text{Total}}$ of items requires QoS restrictions. We will seek to schedule the broadcast of the remaining items in a way that minimizes the total mean waiting time, while upholding all existing constraints.

The reasons behind this approach are proactive QoS and overall efficiency. We assume that a client will probably not abandon a query while waiting for an item without QoS restrictions. Notice, however, that this statement will hold probabilistically.

Recall from Section IV that in push systems, the QoS level is set per item after statistical processing the general preferences of the clients that require it. This processing includes the definition of the item set without QoS requirements as well. In other words, should we delay the transmission of these items or stop broadcasting them completely, a portion of the clients could still react, potentially abandoning the system or requiring the formation of an additional QoS agreement at best. However, as will be proven next (Lemma 3-parameter B), the greater the number of QoS agreements, the more difficult it becomes to uphold them efficiently. Thus, minimizing the overall mean waiting time becomes a form of proactive QoS that favors the stability of the system. Besides, being able to minimize the mean waiting time while upholding all existing QoS requirements adds to the overall efficiency.

Consider a constraint of the form $w_i^{\max} = W_i$. As is evident from Fig. 1, the maximum waiting time $w_i^{\max}$ of an item i, the number of its occurrences in the schedule v_i, and the total size of the schedule L are connected through (2). Thus, setting an item's maximum waiting time ($w_i^{\max}$) via a constraint imposes a restriction of the form

$$L = v_i \cdot w_i^{\max}, \quad i \in S_{\mathrm{QoS}}. \tag{14}$$

The objective is to minimize the global mean waiting time $E_p[w_i]$ [see (3)] with respect to the restrictions (14). The conclusions of the minimization process are summarized as follows.

Lemma 3: Concerning the items with no constraints, the square root rule of (4) minimizes the global mean waiting time $E_p[w_i]$, even when other items with constraints are present.

When constraints are present, the lower bound of $E_p[w_i]$ (5) is transformed as

$$E_p^{'}[w_i]_{\min} = \frac{\left(\sum_{i \notin S_{\mathrm{QoS}}} \sqrt{p_i \cdot l_i}\right)^2}{2 \cdot (1 - B)} + \frac{1}{2} \sum_{i \in S_{\mathrm{QoS}}} p_i \cdot w_i^{\max} \tag{15}$$

where

$$B = \sum_{i \in S_{\mathrm{QoS}}} \frac{l_i}{w_i^{\max}}. \tag{16}$$

It must hold that $B \in [0, 1)$.

Proof: See Appendix B. ∎

The restriction $B \in [0, 1)$ is the mathematical expression of the fact that in a communal system (such as data broadcasting), overly favoring a limited client subset may cause the malfunctioning of the entire system. Consider the example where the maximum waiting time of an item i is set to zero, i.e., $w_i^{\max} = 0$ ($B \to \infty$). Through Fig. 1, it is easy to see that this constraint can be upheld only if the system stopped the broadcasting of all items but i.

Abiding by the restriction of $B \in [0, 1)$ is not really a technical problem, but rather a matter of the degree of democracy in the system. Notice that the final per-item maximum waiting time $w_i^{\max}$ is set centrally by the scheduling authority. If the authority enforces its decisions to the clients unquestionably (no democracy), keeping $B \in [0, 1)$ is trivial. If the authority has no personal motives (maximum democracy), the per-item constraints could be deduced, e.g., through lengthy negotiations among the clients. The matter is strictly context dependent, going beyond the scope of this technical work. For the remainder of this paper, it is assumed that the per-item constraints have been derived in accordance with the restriction $B \in [0, 1)$.

C. Measuring the Impact of Constraints on the System's Performance

Meeting item-specific constraints is the primary goal of a QoS-aware data broadcast scheduling scheme. In the real world, however, upholding constraints come at a price. A group of clients, for example, may ask for more frequent broadcast of sports programs in a TV channel, in exchange for a raised subscription fee. This simple observation highlights two issues.

1) A *reference point* regarding the broadcast frequency of items must be established.
2) A measure of the *impact* of altering the broadcast frequency of an item on the system is needed. This can be used, e.g., to define the final subscription fee. (*Notice, however, that connection with real monetary cost is context dependent and beyond the scope of this paper*).

The *reference point* can be defined as the case of total absence of constraints. The scheduling authority is then simply interested in minimizing the global mean waiting time. This is accomplished by setting the items' broadcast frequencies through (4) (*square root rule*), which leads to the lower bound of (5).

According to Proposition 2, forming a constraint for an item inherently sets its maximum waiting time $w_i^{\max}$. Forcing $w_i^{\max}$ to be less than the value stated by the reference point leaves less room for broadcasting other items (see Fig. 1) and, therefore, increases their individual waiting time. Thus, the *performance impact* of the constraints can be measured by the incurred increase in the waiting time for items $i \notin S_{\mathrm{QoS}}$.

Lemma 4: Assume a set of constraints and the corresponding B value [see (16)]. Furthermore, let

$$\mathrm{IMPACT}(B) = \frac{1 - B_0}{1 - B} \in [1 - B_0, \infty) \tag{17}$$

where $B_0 \in (0, 1)$ is the value of B when the constraints coincide with their reference values:

$$B_0 = 1 - \frac{\sum_{i \notin S_{\mathrm{QoS}}} \sqrt{p_i \cdot l_i}}{\sum_{i \in S_{\mathrm{Total}}} \sqrt{p_i \cdot l_i}}. \tag{18}$$

The mean waiting time $E_u^{'}[w_i]$ for an item $i \notin S_{\mathrm{QoS}}$ is related to its reference point, i.e., $E_u[w_i]$, by

$$\frac{E_u^{'}[w_i]}{E_u[w_i]} = \mathrm{IMPACT}(B). \tag{19}$$

1) The mean waiting time $E_p^{'}[w_i]|_{i \notin S_{\mathrm{QoS}}}$ over all items $i \notin S_{\mathrm{QoS}}$ is also related to its reference value $E_p[w_i]|_{i \notin S_{\mathrm{QoS}}}$ by (19).

2) The total mean waiting time in the case of constraints, i.e., $E_p^{'}[w_i]_{\min}$ [see (15)], is related to its reference value, i.e., $E_p[w_i]_{\min}$ [see (5)], by the approximation:

$$\frac{E_p^{'}[w_i]_{\min}}{E_p[w_i]_{\min}} \approx (\mathrm{IMPACT}(B))^{\gamma} \tag{20}$$

Algorithm 1 The novel QoS-aware Scheduling Algorithm (QoSA)

INPUT

* A set of items $S_{Total} : \{p_i, l_i\}; i = 1 \dots N$,
* A set of constraints for items $i \in S_{QoS}, S_{QoS} \subseteq S_{Total}$, in any Proposition 2-compatible Format.

OUTPUT

* The broadcast schedule that meets the given constraints and minimizes the global mean waiting time,
* The $IMPACT$ metric of the constraints.

1: Translate constraints to $w_i^{\max} = W_i$ Format (1), through Proposition 2.
2: Calculate B via equation (16).
3: **Ensure** $B \in [0, 1)$.
4: Calculate $IMPACT(B)$ via equation (17).
5: Set $v_i = \propto \sqrt{\frac{p_i}{l_i}}, i \notin S_{QoS}$.
6: Calculate $L = \frac{1}{1+B} \sum_{i \notin S_{QoS}} v_i \cdot l_i$.
7: Calculate $v_i, i \in S_{QoS}$ via equation (14).
8: Pass $v_i, i \in S_{Total}$ to the scheduling algorithm of [16] and obtain the returned final schedule.

when $B \in [B_0, 1)$. The parameter $\gamma > 0$ is numerically defined (see Appendix C).

Proof: See Appendix C. ∎

The value B_0 of the B parameter represents the *fair-share (or reference)* point of a wireless broadcast scheduling system. When the constraints yield $B = B_0$, the items get the air time that corresponds to their popularity and size [see (4)]. In practice, values $B > B_0$ mean that a group of clients has paid for favoring the broadcast of certain items. Values $B < B_0$ mean that for a certain price, a set of items have had their broadcast frequencies deliberately lowered (uncommon). Thus, the parameter B expresses the influence of external factors on the system. According to Lemma 4, this impact can be effectively expressed through the metric of (17).

The IMPACT metric expresses the impact of constraints on the mean waiting time per item, per subset of items, or per all items (Lemma 4). Being an increasing function of $B \in [0, 1)$, higher IMPACT values correspond to higher mean waiting time for the items $i \notin S_{\mathrm{QoS}}$.

D. Formulation of the Proposed Quality-of-Service-Aware Scheduling Scheme

The conclusions of Sections IV-A–IV-C are gathered and formulated as the QoSA Algorithm 1.

The algorithm simply produces the QoS-aware number of item instances v_i, which are required to produce the broadcast schedule via the generic serialization process of [16]. In step 5, the $v_i, i \notin S_{\mathrm{QoS}}$ are set via the square root rule of (4). Any multiplicative factor (e.g., 10^{10}) can be used to ensure that $v_i \geq 1, \forall i$, when rounding is applied. (Notice that v_i, $\forall i$, must be integers since they represent occurrences of items in the final schedule.) Finally, step 6 employs a trivial relation between L and B [produced in Appendix B; see (28)]. The algorithm has the same complexity as the serializer of [16], i.e., O(N).

As discussed in Lemma 4, the produced $\mathrm{IMPACT}(B)$ value physically denotes the performance degradation when the targeted QoS restrictions are upheld. In addition, the value $B \in [0, 1)$ shows how difficult it will be to add more QoS restrictions in the future. The further B draws away from the reference point B_0 and approaches the unit, the more difficult it will be to serve additional QoS requirements. Thus, the service provider can utilize the produced IMPACT and B values in order to define the pricing for upholding the current set of restrictions. The clients can then confirm whether they approve the pricing or not, deducing the final set of restrictions after one or several negotiation cycles.

V. Experimental Evaluation of the Quality-of-Service-Aware Scheduling Algorithm

The evaluation of QoSA is divided into three parts:

1) further validation of Rule 2, i.e., the equivalence of the conversion from any metric-specific item constraints to the $w_i^{\max} = W_i$ format;
2) comparison with related work [13], [14] in terms of efficiency in meeting item-specific constraints;
3) further validation of Lemmas 3 and 4, concerning the new lower bound for the global mean waiting time [see (15)] and the measuring of the IMPACT of the constraints on the overall system.

Concerning the nature of the related algorithms, it is noted the *Impatience* and *Jitter* schedulers of [13], [14] target the optimization of system-wide metrics. The minimization of the mean impatience and the jitter overall clients are the set goals. These algorithms, although optimal in their specialization, represent the trend of the related work to handle QoS in push systems in the traditional per-client manner. Thus, item-specific QoS has been overlooked, even though it may be more affinitive toward push systems.

The evaluation is based on the simulation environment of [16]–[18], implemented in MATLAB [28]. The topology comprises a central scheduling and broadcasting server and a number of 1000 wireless clients. The number is purposefully large enough for the central limit theorem to apply on the totality of the client's queries safely. All clients are in the range of the server and wireless propagation and noise issues are idealized in order to focus on the evaluation of the proposed scheduler [1], [12]–[18]. By the same convention, time is measured in Bytes, and conversion to seconds requires the definition of a data transfer rate. Since physical layer issues are idealized, the reader is encouraged to adopt any data transfer rate that corresponds to a physical system of interest.

We consider a number of $N = 100$ data items with sizes $l_i \in [10^{-3}, 10]$ MB, which correspond to typical web-style data sites containing text, images, sound, and animation. The items' request probabilities p_i are set according to the ZIPF distribution, as in the majority of the cited studies. The parameter θ is an input that denotes the skewness parameter of the ZIPF distribution. All runs in this section were repeated for

$\theta \in [0.1 : 0.1 : 1.5]$ but produced identical results. Therefore, all presented figures correspond to $\theta = 0.5$ by convention.

Every simulation run is initialized as follows: A number of 300 000 item queries are generated according to the ZIPF distribution with the given θ value. These are assigned to 1000 clients (300 per client). A client may have one request pending at any given time (*single-threaded clients*). This incurs no loss of generality, since a client with m simultaneous queries can be substituted by m single-threaded clients. For a given set of constraints, Algorithm 1 constructs the broadcast schedule and transmits it periodically to the clients. If a broadcasted item i coincides with the current query of a client, the query is discarded and the client moves to the next query after a random *ThinkTime* of $[0,\ L]$ MB. The latter serves the purpose of taking into account the time a human user requires to process the data before defining the next query. The simulation ends with the depletion of all client queries. Finally, each simulation is repeated 10^2 times in order to extract dependable results.

Random mixed-model constraints are assumed, which employ the following.

1) The incurred impatience metric $I(w_i)$ of [14]:

$$I(w_i) = \begin{cases} 0.5 \cdot \beta_i (w_i - R)^2 w_i, & w_i \geq R_i \\ 0, & w_i < R_i. \end{cases} \tag{21}$$

2) The utility metric of [15]

$$U(w_i) = \begin{cases} U_i^0 \cdot e^{\frac{ln\,2 \cdot (w_i - R_i)}{R_i}}, & w_i \leq R_i \\ U_i^0, & w_i > R_i. \end{cases} \tag{22}$$

3) An arbitrary metric representing scaled service intolerance:

$$C(w_i) = \begin{cases} 1, & w_i < W_1 \\ 3, & W_1 \leq w_i \leq W_2 \\ 100, & w_i > W_2. \end{cases} \tag{23}$$

Illustrations of the impatience and utility metrics are given in Figs. 2 and 3. The clients grow more impatience as the delivery of an item is delayed. In a similar manner, the clients consider an item to be useful if it is delivered in time. Notice that the units of impatience and utility are arbitrary and will be referred to simply as *impatience* and *utility* units. This convention stems from that fact that the metrics $I(w_i)$ [14] and $U(w_i)$ [15] are not strictly defined through medical or statistical research [14], [15]. The reader can infer the physical meaning of these units by mapping them directly to waiting time values in Figs. 2 and 3. Finally, we introduce the metric of scaled service intolerance to demonstrate the flexibility of the QoSA scheme in handling custom metrics as well. We assume that for several items, the clients have been observed to exhibit levels of intolerance as the delivery of a required item is delayed. This behavior is emulated by the scalar form of (23). A value of 100 *intolerance* units indicates the highest degree of client dissatisfaction.

The aforementioned metrics are configured and assigned to the available data items. As shown in Figs. 2 and 3, the impatience and utility metrics change significantly after a given threshold of the waiting time. The same principle applies to the intolerance metric of (23). In order to avoid unrealistically

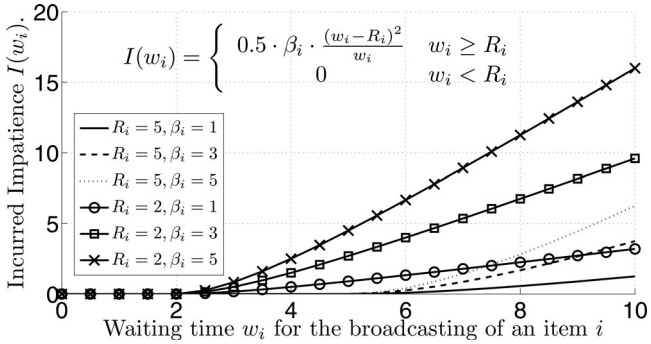


Fig. 2. Form of the impatience metric [14]. The impatience for every item increases as its broadcast is delayed. Two parameters, i.e., β_i and R_i, are used to customize the form of the plot.

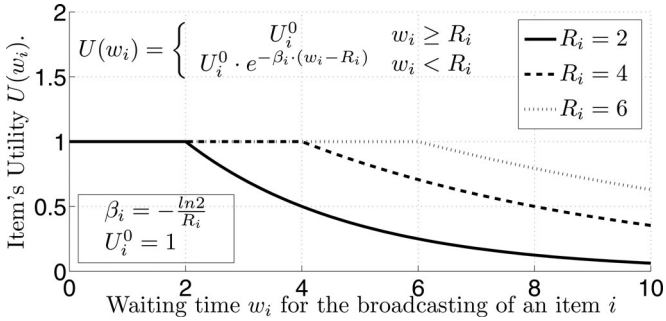


Fig. 3. Form of the utility metric [15]. The utility of any data item drops as its broadcast is delayed. R_i, β_i, and U_i^0 are parameters that regulate the form of the plot.

small threshold values (e.g., not willing to wait for the duration of a single item), we set this parameter as a portion of the total item duration, $D = \sum_{i=1}^{N} l_i$. The exact parameter values and assignments to items are as follows.

1) For items $i = 1, \ldots, 10$, the incurred impatience is of primary concern, with parameter values $R_i = 10\% \cdot D$ and $\beta_i = 1$. This customization is denoted as $I_1(w_i)$.
2) Items $i = 11, \ldots, 20$ are assigned the $I_2(w_i) : \{R_i = 30\% \cdot D,\ \beta_i = 5\}$ customization.
3) Items $i = 21, \ldots, 30$ employ utility metric, using the configuration $U_1(w_i) : \{R_i = 20\% \cdot D,\ U_i^o = 50\}$.
4) Items $i = 31, \ldots, 40$ are assigned to $U_2(w_i) : \{R_i = 15\% \cdot D,\ U_i^o = 20\}$.
5) Items $i = 41, \ldots, 50$ use the intolerance metric C of (23), for $W_1 = 5\% \cdot D$ and $W_2 = 15\% \cdot D$.
6) Finally, items $i = 51, \ldots, 100$ have no associated metric or constraint and are of no interest to the comparison.

While the aforementioned assignment is random, it produces results representative of all other tested cases.

We proceed to set a constraint for each item, $i = 1, \ldots, 50$, based on the mean value of its associated metric (e.g., $E_u[I_1(w_1)] \leq 51$). This way of expression corresponds to *Format* 3 of Rule 2. The targeted mean values are randomly picked from the range 1, 200. The scheduling algorithms of [13] and [14] and the QoSA scheme are then requested to produce a schedule that upholds these constraints. The results are shown in Fig. 4.

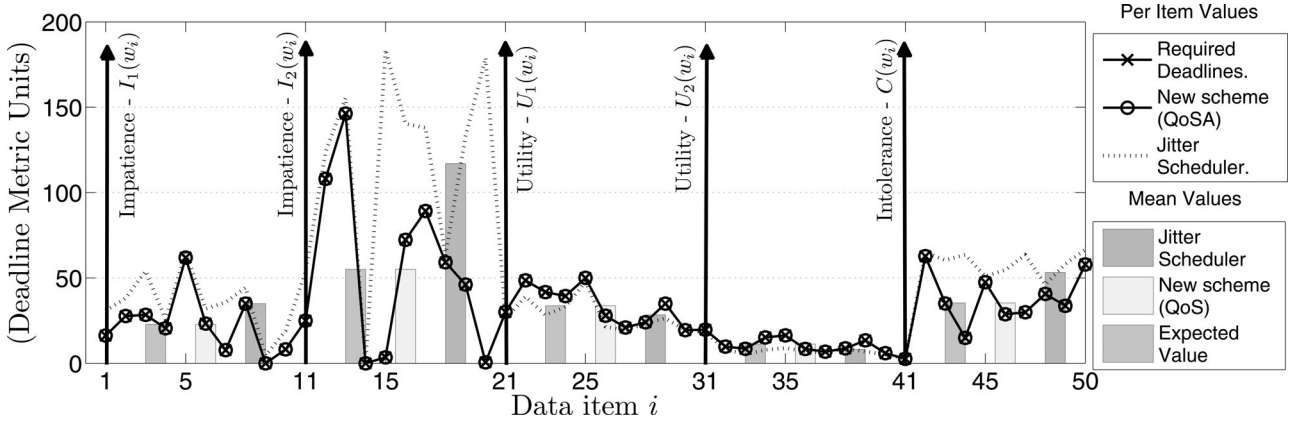


Fig. 4. Performance comparison of the proposed algorithm (QoSA) and the Jitter Scheduler of [13]. Closest match with the plot expressing the required constraints indicates better performance. QoSA upholds all constraints with near-perfect accuracy, while achieving better overall mean values as well.

The proposed algorithm matches all constraints accurately. The results also abide by the translation Rules 1 and 2, since internally QoSA translates all constraints to *Format*1 (see Section IV-D, Algorithm 1, Step 1). The algorithm in [13] produces a schedule that generally incurs more impatience and less utility in the majority of the cases, while no constraint is reliably met. Furthermore, as discussed in [13], this algorithm requires an arbitrary parameter, $\alpha \in [2, 3]$, as input, which can be defined in no other way but through brute force. The simulation was repeated for all values $\alpha = 2 : 0.05 : 3$ and the schedule that produced the lowest mean square error with regard to the required constraints was kept ($\alpha = 2.15$). The algorithm in [14] did not produce results for this configuration. This algorithm states that the item to be broadcasted at a time moment t is the one maximizing

$$F_i = (t - T_i).^2 \cdot \frac{p_i}{l_i} \cdot \left.\frac{\partial M_i}{\partial w_i}\right|_{w_i = t - T_i} \tag{24}$$

where T_i is the time moment when item i was last broadcasted. It is obvious that for the utility metric $U(w_i)$, the first derivative approaches zero as w_i increases. The derivative of the custom metric $C(w_i)$ [see (23)] is also zero (apart from just two discrete cases). The same applies to items with no constraint. The algorithm in [14], thus, entered an infinite loop broadcasting only items with associated impatience $I(w_i)$ metrics. To enable comparison with [14], we assign the impatience metric $I_1^{'}(w_i) : \{R_i = 10\% \cdot D, \beta_i = 1\}$ to items $i = 1, \ldots, 50$. Absence of constraints for items $i = 51, \ldots, 100$ is emulated by assigning the metric $I_2^{'}(w_i) : \{R_i = 10\% \cdot D, \beta_i = 0.1\}$, (i.e., a much lower β value).

The results are given in Fig. 5. The proposed algorithm satisfies accurately all constraints, while the algorithm in [14] targets only the minimization of the global mean impatience. These issues do not lessen the importance of [13] and [14], which pioneered in introducing QoS in periodic scheduling. However, they highlight the need for flexible QoS schemes.

The remainder of the experimental evaluation focuses on studying the IMPACT metric and examining the main points of the mathematical analysis further. For the same configuration leading to Fig. 4, all constraints are scaled by a common

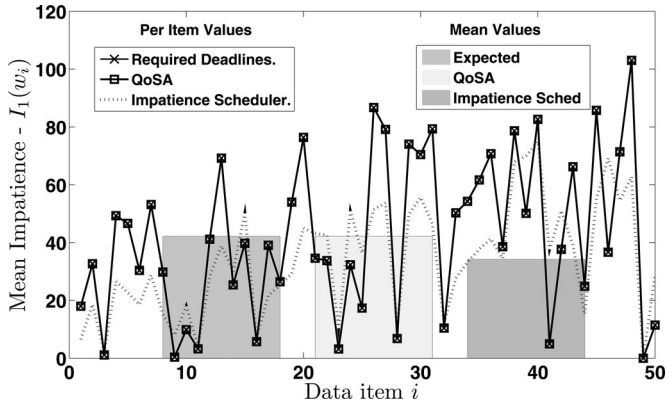


Fig. 5. Performance comparison of QoSA and the Impatience Scheduler of [14]. The proposed algorithm upholds all constraints with near-perfect accuracy. The impatience scheduler achieves better mean impatience, but cannot uphold individual, per-item constraints.

factor in order to produce parameter values $B = 0.1 : 0.1 : 0.9$ [see (16)]. Each time, the global mean waiting time achieved by QoSA is logged and compared with the analytical lower bound for (15). As shown in Fig. 6(a), the values coincide, matching the analysis. The procedure is repeated, this time logging the IMPACT of the constraints on the mean waiting time for items with no associated constraint. The results in Fig. 6(b) show that the simulation-derived values coincide with those projected by the first two assertions of Proposition 2, essentially matching their analytical proof (see Appendix C). Fig. 6(c) illustrates the simulation-derived and the analysis-derived IMPACT of the presence of agreements on the total mean waiting time. Notice that the approximation of (20) is shown to be adequate for $B \in [B_0, 1)$ as expected. The curve-fitting parameter γ was derived by applying mean square error-based curve fitting. Notice that the cases, where $B \in [B_0, 1)$, correspond to constraints demanding smaller waiting time for an item than its fair-share. Intuitively, this is expected to be the most common and interesting case.

An alteration of (20) can be defined, if values $B \in [0, B_0]$ are of interest as well. Noticing that the global mean waiting time is driven to infinity when $B \to 0$ (i.e., when one or more items

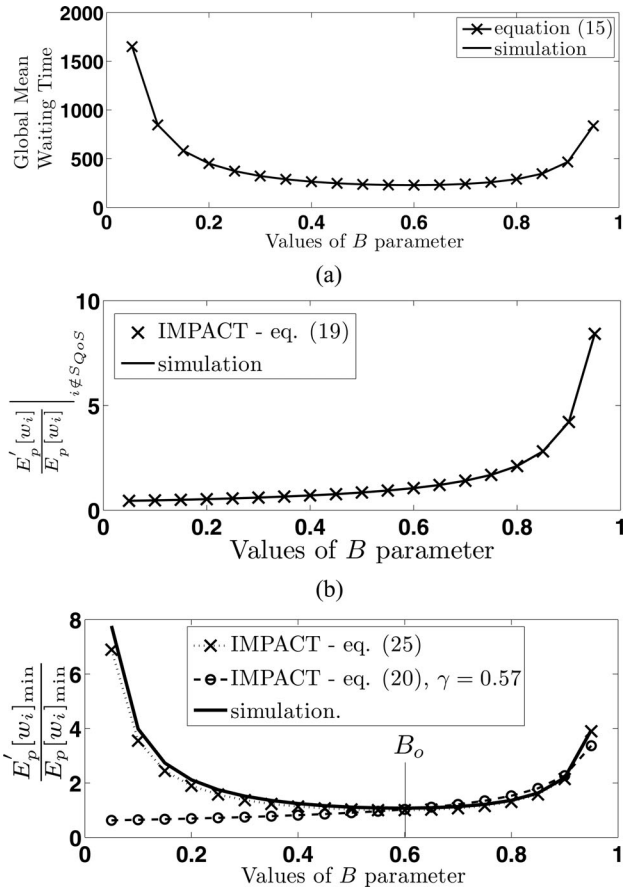


Fig. 6. Study through simulation of the IMPACT metric and the main points of the mathematical analysis. The analysis-derived values nearly coincide with simulation-derived ones in all cases. (a) Comparison of expected and resulting global mean waiting time. (b) Comparison of expected and resulting IMPACT of existing constraints on the mean waiting time for items with no associated constraints. (c) Comparison of expected and resulting IMPACT of existing constraints on the global mean waiting time.

stop being broadcasted although several clients still need them), we propose the following approximation:

$$\frac{E_p'[w_i]_{\min}}{E_p[w_i]_{\min}} \approx (1-B_0)\frac{1-Bo}{1-B} + B_0\frac{Bo}{B}. \tag{25}$$

(This approximation also satisfies the boundary conditions described in Appendix C.) Fig. 6(c) shows (25) to match the simulation-derived results. The same conclusion was repeated for any of the tested simulation configurations. Equation (25) can serve as an efficient metric of the IMPACT of constraints, when we are solely interested in keeping the total mean waiting time in check.

VI. Conclusion

A novel QoS-aware broadcast scheduling scheme for wireless broadcast systems has been presented. Two major omissions of the related work were the overspecialization of QoS-related metrics and the lack of potential to keep item-specific constraints. The proposed algorithm has been proven to be QoS metric independent. It manages to uphold any data-item-specific constraint accurately, while minimizing the global mean waiting time. Finally, accurate analytical metrics that quantify the impact of QoS agreements on the overall performance of the system were defined.

Appendix A
Proof of Proposition 2

The first case is a direct result of the fact that $M_i = M_i(w_i)$ is an n:1 mapping of $w_i \to M_i$: For a given $M_i^{\max}$, there are always one or more corresponding w_i values. Choosing the maximum of them ensures that more air time is given for other items (see Fig. 1), while still upholding the constraint. The same applies to case 2. Concerning the third case, the mean value of M_i is

$$E_u[M(w_i)] = \int_0^{w_i^{\max}} M_i(w_i)\cdot P(w_i)\cdot dw_i. \tag{26}$$

Since u is uniform, $P(w_i) = 1/w_i^{\max}, \forall i$. Then

$$E_u[M(w_i)] = \frac{\int_0^{w_i^{\max}} M_i(w_i)\cdot dw_i}{w_i^{\max}}. \tag{27}$$

The maximum value of $w_i^{\max}$ that satisfies (27) is chosen, as in the first two cases. Since $w_i \in [0, w_i^{\max}]$ uniformly, it holds that $E_u[w_i] = 0.5\cdot w_i^{\max}$. Q.E.D.

Appendix B
Proof of Lemma 3

We insert restrictions (14) in (1), replacing the $v_i, i \in S_{\text{QoS}}$. Notice that L will then be present in both parts of the equation. Solving for L produces

$$L = \frac{\sum_{i\notin S_{\text{QoS}}} v_i\cdot l_i}{1-\sum_{i\in S_{\text{QoS}}}\frac{l_i}{w_i^{\max}}} \overset{\text{equation(16)}}{=} \frac{\sum_{i\notin S_{\text{QoS}}} v_i\cdot l_i}{1-B}. \tag{28}$$

Since L is the length of the schedule, it must hold $L > 0 \Leftrightarrow B < 1$. B is also an inherently positive quantity; hence, the restriction $B \in [0,1)$. For an arbitrary item $j \notin S_{\text{QoS}}$ and a given L, (28) yields

$$\frac{\partial v_j}{\partial v_i} = -\frac{l_i}{l_j}. \tag{29}$$

The objective is to find the $v_i^o, i \notin S_{\text{QoS}}$, that minimize (3). Requesting $\frac{\partial E_p[w_i]}{\partial v_i} = 0$ produces through (29)

$$v_i^o = \left(v_j\cdot\sqrt{\frac{l_j}{p_j}}\right)\cdot\sqrt{\frac{p_i}{l_i}}, \qquad \forall i \notin S_{\text{QoS}} \tag{30}$$

i.e., $v_i^o \sim \sqrt{\frac{p_i}{l_i}}$. Finally, (15) is produced by inserting (28) and (30) in (3), after trivial calculations.

Appendix C
Proof of Lemma 4

The ($'$) notation will be employed for denoting the presence of constraints. The mean waiting time of an item $i \notin S_{\text{QoS}}$ is

$$E_u'[w_i] = \frac{(w_i^{\max})'}{2} \overset{\text{equation(2)}}{=} \frac{L'}{2\cdot v_i^o}. \tag{31}$$

When constraints are present, L' is given by (28), and $v_i^o \sim \sqrt{\frac{p_i}{l_i}}$ in any case. Equation (31), thus, becomes

$$E_u^{'}[w_i] = \left(\sum\nolimits_{i \notin S_{\mathrm{QoS}}} \sqrt{p_i \cdot l_i}\right)\left(2 \cdot (1-B) \cdot \sqrt{\frac{p_i}{l_i}}\right). \quad (32)$$

For the reference point (no constraints), L is given by (1), and once again $v_i^o \sim \sqrt{\frac{p_i}{l_i}}$. Therefore

$$E_u[w_i] = \frac{L}{2 \cdot v_i^o} = \left(\sum\nolimits_{i \in S_{\mathrm{Total}}} \sqrt{p_i \cdot l_i}\right)\left(2 \cdot \sqrt{\frac{p_i}{l_i}}\right). \quad (33)$$

Equation (19) is derived by dividing (32) by (33), and setting

$$B_0 = 1 - \frac{\sum_{i \notin S_{\mathrm{QoS}}} \sqrt{p_i \cdot l_i}}{\sum_{i \in S_{\mathrm{Total}}} \sqrt{p_i \cdot l_i}}, \quad B_0 \in [0,1]. \quad (34)$$

In a straightforward manner, concerning the mean over the waiting time for all $i \notin S_{\mathrm{QoS}}$

$$\left.\frac{E_p^{'}[w_i]}{E_p[w_i]}\right|_{i \notin S_{\mathrm{QoS}}} = \frac{\sum_{i \notin S_{\mathrm{QoS}}} p_i \cdot E_u^{'}[w_i]}{\sum_{i \notin S_{\mathrm{QoS}}} p_i \cdot E_u[w_i]} \overset{\text{equation(19)}}{\Rightarrow}$$

$$\left.\frac{E_p^{'}[w_i]}{E_p[w_i]}\right|_{i \notin S_{\mathrm{QoS}}} = \left(\frac{1-B_0}{1-B}\right)\frac{\sum_{i \notin S_{\mathrm{QoS}}} p_i \cdot E_u[w_i]}{\sum_{i \notin S_{\mathrm{QoS}}} p_i \cdot E_u[w_i]}$$

$$= \left(\frac{1-B_0}{1-B}\right). \quad (35)$$

For the third part of Lemma 4, we divide (15) by (5), producing

$$\frac{E_p^{'}[w_i]_{\min}}{E_p[w_i]_{\min}} = \frac{(1-B_0)^2}{1-B} + \frac{\sum_{i \in S_{\mathrm{QoS}}} p_i \cdot w_i^{\max}}{\left(\sum_{i \in S_{\mathrm{Total}}} \sqrt{p_i \cdot l_i}\right)^2}. \quad (36)$$

For the bound $B = B_0$, (36) yields $\frac{E_p^{'}[w_i]_{\min}}{E_p[w_i]_{\min}} = 1$, and for $B \to 1$, $\lim_{B \to 1} \frac{E_p^{'}[w_i]_{\min}}{E_p[w_i]_{\min}} \to \infty$. Furthermore, (36) is an increasing function of $1/(1-B)$. This constitutes (20) a good candidate for performing curve fitting of (36) for $B \in [B_0, 1)$, since it also respects the two boundary conditions. The $\gamma > 0$ is the curve-fitting parameter, which can be numerically calculated by any popular method, such as the minimization of the mean square error.

REFERENCES

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

[2] J. Xu, X. Tang, and W.-C. Lee, "Time-critical on-demand data broadcast: Algorithms, analysis, and performance evaluation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 1, pp. 3–14, Jan. 2006.

[3] C. Ngo and V. Li, "Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1439–1441, Sep. 2003.

[4] Y. Yang, Z. Cao, M. Tan, and J. Yi, "Fairness and dynamic flow control in both unicast and multicast architecture networks," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 37, no. 2, pp. 206–212, Mar. 2007.

[5] J. Cao, X. Feng, J. Lu, H. Chan, and S. Das, "Reliable message delivery for mobile agents: Push or pull?," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 5, pp. 577–587, Sep. 2004.

[6] M. d. Bruijne, "Next generation critical infrastructures: The push and pull to real-time," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2004, pp. 4655–4661.

[7] A. Celik, A. Datta, and S. Narasimhan, "Supporting subscription oriented information commerce in a push-based environment," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 30, no. 4, pp. 433–445, Jul. 2000.

[8] V. L. Kakali, P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "Speeding up the adaptation process in adaptive wireless push systems by applying spline interpolation technique," in *Proc. 15th IEEE Mediterranean Electrotech. Conf.*, Valletta, Malta, Apr. 2010, pp. 1263–1268.

[9] M. Caprioglio, T. Planterose, and C. Morel, "An end-to-end architecture for advanced multimedia broadcast," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2001, pp. 3217–3222.

[10] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 797–819, Nov. 2011.

[11] W.-N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 29–43, Jan. 2009.

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

[13] S. Jiang and N. H. Vaidya, "Response time in data broadcast systems: mean, variance and tradeoff," *Mobile Netw. Appl.*, vol. 7, no. 1, pp. 37–47, 2002.

[14] M. Raissi-Dehkordi and J. S. Baras, "Broadcast scheduling for time-constrained information delivery," in *Proc. IEEE Global Telecommun. Conf.*, Washington, DC, Nov. 2007, pp. 5298–5303.

[15] R. Dewri, I. Ray, I. Ray, and D. Whitley, "Optimizing on-demand data broadcast scheduling in pervasive environments," in *Proc. 11th Int. Conf. Extending Database Technol.*, Nantes, France, Mar. 2008, pp. 559–569.

[16] C. Liaskos, S. Petridou, and G. Papadimitriou, "Towards realizable, low-cost broadcast systems for dynamic environments," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 383–392, Apr. 2011.

[17] C. Liaskos, S. Petridou, G. Papadimitriou, P. Nicopolitidis, M. Obaidat, and A. Pomportsis, "Clustering-driven wireless data broadcasting," *IEEE Wireless Comm. Mag.*, vol. 16, no. 6, pp. 80–87, Dec. 2009.

[18] C. Liaskos, S. Petridou, and G. Papadimitriou, "Cost-aware wireless data broadcasting," *IEEE Trans. Broadcasting*, vol. 56, no. 1, pp. 66–76, Mar. 2010.

[19] W. Rouse, "Need to know-information, knowledge, and decision making," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 4, pp. 282–292, Nov. 2002.

[20] 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," presented at the 15th Int. Conf. Embedded Real-Time Comput. Syst. Appl., Beijing, China, Aug. 2009.

[21] S. H. Kang, S. Choi, S. J. Choi, G. Lee, J. Lew, and J. Lee, "Scheduling data broadcast based on multi-frequency in mobile interactive broadcasting," *IEEE Trans. Broadcasting*, vol. 53, no. 1, pp. 405–411, Mar. 2007.

[22] S. Kim and S. H. Kang, "Scheduling data broadcast: An efficient cut-off point between periodic and on-demand data," *IEEE Commun. Lett.*, vol. 14, no. 12, pp. 1176–1178, Dec. 2010.

[23] U. Balli, H. Wu, B. Ravindran, J. Anderson, and E. Jensen, "Utility accrual real-time scheduling under variable cost functions," *IEEE Trans. Comput.*, vol. 56, no. 3, pp. 385–401, Mar. 2007.

[24] V. C. S. Lee, X. Wu, and J. K.-Y. Ng, "Scheduling real-time requests in on-demand data broadcast environments," *Real-Time Syst.*, vol. 34, no. 2, pp. 83–99, 2006.

[25] B. Ravindran, E. Jensen, and P. Li, "On recent advances in time/utility function real-time scheduling and resource management," in *Proc. 8th IEEE Int. Symp. Object-Oriented Real-Time Distrib. Comput.*, Seattle, WA, May 2005, pp. 55–60.

[26] J. Grossklags and C. Schmidt, "Software agents and market (in) efficiency: A human trader experiment," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 1, pp. 56–67, Jan. 2006.

[27] C. Li, J. Giampapa, and K. Sycara, "Bilateral negotiation decisions with uncertain dynamic outside options," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 1, pp. 31–44, Jan. 2006.

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

Authors' photographs and biographies not available at the time of publication.