Parallel Data Broadcasting for Optimal Client Service Ratio

Christos Liaskos, *Graduate Member, IEEE,* Georgios Papadimitriou, *Senior Member, IEEE,*
Petros Nicopolitidis, *Senior Member, IEEE,* and Andreas Pomportsis

***Abstract*—This letter introduces service ratio-based optimization in multichannel data broadcast systems. The classic pursuit of data broadcast scheduling has revolved around QoS metrics, such as the minimization of the mean serving time or jitter. Recent studies in single channel systems have shown that this approach reduces the client service ratio, essentially forcing clients to abandon the system. Content discarding in single channel systems was then proven to maximize the service ratio. The present work introduces a novel algorithm for service-ratio-oriented assignment of data to multiple channels with different bandwidths. Comparison to related approaches demonstrates the superior scalability of the proposed algorithm.**

***Index Terms*—Multichannel broadcasting, push systems, service ratio.**

I. Introduction

PUSH-BASED broadcasting offers increased bandwidth efficiency and unlimited scalability [1]. As in other communal settings [2], cognitive scheduling is required for covering the client needs in an optimal way. Landmark and recent studies concluded that periodic data item transmissions optimize the QoS, typically expressed in the form of the mean query serving time [3], jitter or energy efficiency [4]. However, optimal QoS was shown not to entail optimal service ratio [5] in single-channel environments. By discarding most of the available data items, the broadcasting authority can get the highest possible service ratio, which is much lower than the ideal 100% nonetheless. The question posed is the following. Given M available channels with relative bandwidths b_j, $j = 1 \dots M$ ($b_M = 1$, $b_j > b_{j+1}$), how should the scheduling authority assign the available data items in order to maximize the service ratio?

The study assumes N data items, $i = 1 \dots N$, with sizes l_i measured in seconds required for transmission in the slowest channel (b_M). Transmission in other channels would then require ${}^{l_i}/_{b_j}$ seconds. Each item is also associated to a request probability, p_i, expressing the percentage of the total system queries that refer to item i. Finally, each item has a metric of criticality expressed by a parameter $c_i > 0$. A client will wait for a random interval picked from the exponential distribution $P_{abandon}(w) = c_i \cdot e^{-c_i \cdot w}$ before abandoning the query. All item attributes, p_i, c_i, l_i, are assumed to be known. The reader is directed to [6] for studies on adaptivity. Each client may have only one pending query at a given moment and no caching strategy (discussed as a separate functionality layer, e.g. [7]). Notice that in push systems client queries are *silent*, i.e. their is no interaction with the scheduling server (as in pull-based broadcasting, e.g. [8]). A client simply listens to the broadcast channels, waiting for time w. An indexing scheme, incorporated to the broadcast schedule, informs the clients of the data-channel assignments [4]. When optimizing QoS or service ratio, a push-based broadcast schedule must be periodic [3], [5]. The same sequence of data items is broadcasted repeatedly. Each item has v_i periodic occurrences in this sequence, derived from its attributes. Scheduling algorithms define the v_i values that correspond to a given optimization criterion. Serialization algorithms then produce the final sequence, using v_i and l_i as inputs [9].

Extending [5], the present study will demonstrate that the inequivalence of QoS and service ratio pursuits applies to multichannel configurations as well. The single-channel content selection scheme of [5] will be transformed into a multichannel, bandwidth-efficient data assignment scheme. The scheme operates in two steps, tweaking an initial, channel-conserving assignment towards a final, fully channel-exploitative solution.

Manuscript received June 18, 2012. The associate editor coordinating the review of this letter and approving it for publication was H. Luo.

The authors are with the Dept. of Informatics, Aristotle University P.O.B. 888, Thessaloniki 54124, Greece (e-mail: {cliaskos, 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 Identifier 10.1109/LCOMM.2012.091212.1339

II. Methodology

The query service ratio is defined as the fraction of not-abandoned queries to the total number of queries posed over an infinite time horizon. The study of [5] on single-channel systems stated that a data set corresponds to maximum service ratio if and only if:

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

where $W(.)$ is the Lambert-W function [10]. If not, the item with the minimal ${}^{p_i}/_{l_i \cdot c_i}$ should be discarded iteratively, until (1) holds. An item with minimal ${}^{p_i}/_{c_i \cdot l_i}$ ratio interests few clients (lower p_i), but has too strict deadlines (high c_i) and bandwidth requirements (high l_i). In other words, serving it would require too many resources with very little gain.

The optimal item occurrences that maximize the client service ratio are given by [5]:

$$v_i^{opt} \propto \frac{-c_i}{1 + W\left(\frac{-p_i + V \cdot l_i \cdot c_i}{e \cdot p_i}\right)} \tag{2}$$

where V is a constant which is calculated numerically from the condition $\sum_{i=1}^{N} v_i^{opt} \cdot l_i = 1$. The expected service ratio is

1089-7798/12$31.00 © 2012 IEEE

then equal to [5]:

$$\mathcal{SR} = \sum_{i=1}^{N} p_i \cdot v_i^{opt} \frac{1 - e^{-\frac{c_i}{v_i^{opt}}}}{c_i} \tag{3}$$

Notice that the above process discards content to become inline with equation (1). Given an original set of data items, (1) will lock on a subset that presents a local maximum. Equation (3) can be used for comparing these local maxima and finding the globally optimal data subset. In addition, the item discarding rule is to remove the item with the minimal $p_i/c_i \cdot l_i$ value. This implies that a content selection process can operate on an ascending ordering of the items by this ratio, reducing the overall complexity. Finally, notice that this iterative discarding of items will result into optimal subsets of maximum size, since discarding stops exactly when equation (1) begins to hold. This approach will be denoted as fully loading the given channel**.**

The above process could be applied repeatedly for a multi-channel configuration. In each iteration, the process locks on the optimal data subset and assigns it to the next free channel with the maximum relative bandwidth, b_j. The channel is then marked as occupied, and the data subset is removed from the original set. The process repeats until the depletion of data items or channels. Notice that this initial stage will seek to fully load each channel, thus minimizing the total number of occupied channels. The stage is thus characterized as channel-conserving and is formulated in Table 1 as PARABOLES-CS.

Maximizing the service ratio means that all channels must be exploited. Utilizing all channels is a design policy, rather than a strict outcome. Using as many channels as possible for data dissemination is a form of parallel processing. It is thus expected to be beneficial in terms of service ratio. However, utilizing a very slow channel may not be beneficial at all. Following this rationale, PARABOLES-CS will hand over operations to PARABOLES-EX, which will then expand to the remaining channels, if this action is beneficial in terms of service ratio.

The PARABOLES-EX stage is given in Table 2. It performs iterative hill-climbing on the solution of the PARABOLES-CS stage, seeking the maximization of the service ratio. In each iteration, every channel offers to offload its previous channel (by the b_j sorting) by taking the item with the minimal $p_i/c_i \cdot l_i$ value, unless it makes itself unoptimizable (eq. (1)). The transaction that yields the maximum profit in terms of projected service ratio (by eq. (3)) is committed. If no profit can be obtained, the algorithm terminates. Thus, the exploitation of the available channels is achieved. Notice that eq. (3) holds for multiple channels, provided that each item size l_i has been divided by the corresponding channel bandwidth, b_j.

Algorithm 1 STAGE 1: Channel-Conserving, Parallel Broadcasting for Optimal Expected Service (PARABOLES-CS).

INPUT: **I)** A data set, $\mathcal{D}$: $\{p_i, c_i, l_i\}$, $i = 1 \ldots N$, sorted by ascending $p_i/l_i \cdot c_i$ value.
II) A number of M data channels sorted by descending relative bandwidth, $\{b_1, b_2, \ldots, b_j, \ldots, b_M = 1\}$
OUTPUT: The optimal, data subset channel assignments, $\mathcal{S}_j$, $j = 1 \ldots M$.

process data_assignments
  for $j = 1 : M$
    $\mathcal{S}_j$ =best_local_subset($\mathcal{D}$, b_j);
    $\mathcal{D} = \mathcal{D} - \mathcal{S}_j$;
  end
return $\mathcal{S}_j$, $j = 1 \ldots M$**;**

process best_local_subset(**an item set** $\mathcal{F}$, bw b)
  $I_1 = 1$; $I_2 = |\mathcal{F}|$;
  Init $best\_subset = []$; $best\_ratio = 0$;
  while $I_2 \geq I_1$
    Subset $\mathbb{I}^* = \mathcal{F}\,(I_1 : 1 : I_2)$, inclusive;
    Set $p_i^* = p_i(\mathbb{I}^*)/\sum p_i(\mathbb{I}^*)$; $l_i^* = l_i(\mathbb{I}^*)/b$; $c_i^* = c_i(\mathbb{I}^*)$;
    if set_is_optimizable($\{p_i^*, l_i^*, c_i^*\}$) **then** /*eq. (1)*/
      Calculate $\mathcal{SR}$ from equation (3);
      if $\mathcal{SR} > best\_rate$ **then**
        Set $best\_rate = \mathcal{SR}$; $best\_subset = \mathbb{I}^*$;
        $I_2 = I_1 - 1$; $I_1 = 1$;
      end if
    else
      $I_1 = I_1 + 1$;
    end if
  end while
return $best\_subset$;

III. Simulations

The proposed algorithm is compared through simulation to the DRP/CDMS scheme of [11]. The opposed scheme hill-climbs an initial solution, called "DRP", transforming it to the "DRP/CDMS". It targets the minimization of the mean query serving time. The goal of the comparison is to conclude if superior QoS yields higher service ratio. Both stages of the compared algorithms are presented, in order to demonstrate the effects of hill-climbing on the initial solutions.

The setup consists of $N = 30$ items with sizes $l_i = 1sec$, $\forall i$ and $c_i = 0.5$, $\forall i$. The p_i values are set by the Zipf distribution ($p_i \propto i^{-\theta}$), as in [11], with skewness factor $\theta = 0.95$. The simulation considers the setup of Section I with 300 clients, each posing 1000 sequential queries before the simulation ends. After an answered or abandoned query, a client waits for a $ThinkTime$ randomly picked from $[1,\,20]sec$. The number M and relative bandwidths b_j of the available channels vary per examined case.

Fig. 1 refers to varying number of channels with equal, unary bandwidth. The PARABOLES algorithm offers significantly better service ratio for small numbers of available channels. Its inherent ability to discard items is proven beneficial in terms of service ratio. As the number of channel increases (and the assignment becomes more trivial) PARABOLES and DRP-CDMS exhibit similar performance. Note that the PARABOLES-EX stage has no effect when the channels are too few to optimally serve all data ($M \leq 5$), as expected. In the case of channels with different bandwidths (Fig. 2), the previous remarks are retained and amplified. The channel bandwidths are set as $b_M = M$, $b_j = b_{j+1} - 1$, $j = (M-1)..1$, i.e. each time, a better channel is introduced to the system con-

Algorithm 2 STAGE 2: Channel-Exploiting, Parallel Broadcasting for Optimal Expected Service (PARABOLES-EX).

INPUT: The channel assignments, $\mathcal{S}_j$, $j = 1 \dots M$, produced by the PARABOLES-CS stage.

OUTPUT: The modified, exploitative assignments, $\mathcal{S}_j^*$, $j = 1 \dots M$.

```
SR^max = projected service ratio; J = NULL;
while (true) do
  SR = 0;
  for j = 1 to M − 1
    Move item S_j(1) to start of S_{j+1};
    SR = projected service ratio; /*eq. (3)*/
    if SR > SR_max then
      SR_max = SR; J = j;
    end if
    Rollback item movement;
  end for
  if SR == 0 then
    break while;
  else
    Move item S_J(1) to S_{J+1};
  end if
end while
return produced S_j as S_j^*, j = 1 . . . M;
```

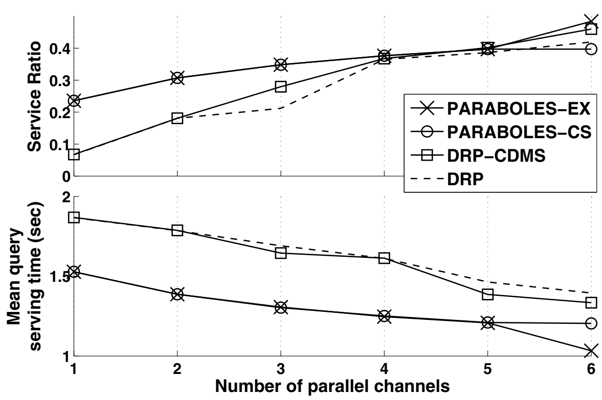

Fig. 1. Achieved service ratios and mean times of the compared schedulers for channels with equal bandwidths. PARABOLES offers better service ratios than DRP in all cases. The demonstrated runs also correspond to a case where dominance in service ratio goes along with smaller service times. This state is reversed in Fig. 2.

figuration. The service ratio advantage of the PARABOLES algorithm continues to hold.

Most importantly, the results show that better service times (QoS) do not mean better service ratio. While generally proportional, the relation between the two quantities is not one-to-one. In Fig. 2 case $M = 5$, for example, DRP surpasses PARABOLES in terms of QoS. However, PARABOLES offers higher service ratio. In other cases, PARABOLES is better in both aspects. In all cases, PARABOLES offers the highest service ratio as expected. Thus, it is shown that client set maximization should be targeted directly in a standalone manner, rather than indirectly, through QoS upgrades.

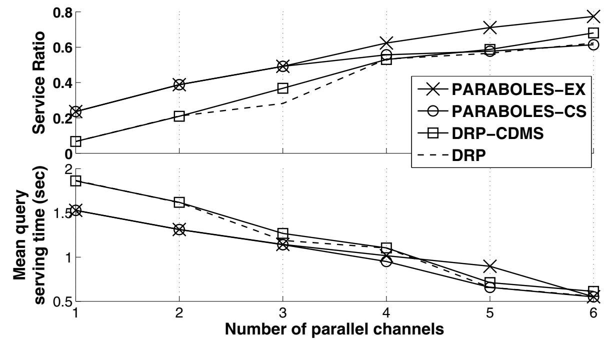

Fig. 2. When using channels with different bandwidths, PARABOLES still offers the highest service ratios. However, DRP may surpass PARABOLES in serving time (QoS).

IV. Conclusion

The present paper demonstrated the significance of directly optimizing the expected service ratio in wireless data broadcast systems. A novel algorithm representing this approach was presented and compared to related solutions prioritizing QoS. The results demonstrated that QoS-centric approaches may lack significantly in terms of achieved service ratio.

References

[1] S. C.-H. Huang, S. Y. Chang, H.-C. Wu, and P.-J. Wan, “Analysis and design of a novel randomized broadcast algorithm for scalable wireless networks in the interference channels,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 7, pp. 2206–2215, 2010.

[2] P. Sarigiannidis, G. Papadimitriou, and A. Pomportsis, “A high performance scheduling priority scheme for WDM star networks,” *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 76–78, 2007.

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

[4] 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.

[5] C. K. Liaskos, A. N. Tsioliaridou, and G. I. Papadimitriou, “More for less: getting more clients by broadcasting less data,” in *Proc. 2012 International Conference on Wired/Wireless Internet Communications*, pp. 64–75.

[6] P. Nicopolitidis, V. L. Kakali, G. I. Papadimitriou, and A. S. Pomportsis, “On performance improvement of wireless push systems via smart antennas,” *IEEE Trans. Commun.*, vol. 60, no. 2, pp. 312–316, 2012.

[7] C. Zhan, V. C. S. Lee, J. Wang, and Y. Xu, “Coding-based data broadcast scheduling in on-demand broadcast,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3774–3783, 2011.

[8] 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, 2010.

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

[10] 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.

[11] H.-P. Tsai, H.-P. Hung, and M.-S. Chen, “On channel allocation for heterogeneous data broadcasting,” *IEEE Trans. Mobile Computing*, vol. 8, no. 5, pp. 694–708, 2009.