Spectrum Load Balancing for Medium Access in Cognitive Radio Systems

A. T. Chronopoulos, *Senior Member, IEEE*, M. R. Musku, *Member, IEEE*,
S. Penmatsa, *Member, IEEE*, and D. C. Popescu, *Senior Member, IEEE*

Abstract— **In this work, the problem of spectrum allocation in cognitive radios is shown to be similar to the load balancing problem in distributed computer systems. Spectrum Load Balancing (SLB) algorithm based on the non-cooperative load balancing problem in computers is proposed, and is applied to a cognitive radio system. The capability of SLB to support QoS in the presence of other competing cognitive networks is evaluated via simulations and compared with the existing Spectrum Load Smoothing (SLS) algorithm. SLB is more efficient than SLS and it provides a Nash Equilibrium.**

Index Terms— **Spectrum allocation, game theory, load balancing.**

I. Introduction

FLEXIBLE and dynamic spectrum usage requires an intelligent medium access, especially in the face of QoS support. Radios designed for efficiently using a shared spectrum and at the same time not causing significant interference to the incumbent (primary license holding) radio systems are referred to as Cognitive (spectrum agile) radios [1]. They identify radio spectrum when it is unused by the incumbent radio system and use this spectrum in an intelligent way based on spectrum observation. A method referred to as Spectrum Load Smoothing (SLS) to coordinate and optimize the usage of radio spectrum is introduced in [2], [3]. SLS uses an opportunistic spectrum usage framework which causes no interference to primary users of the spectrum (either because the primary user is not currently active, or because it is unlicensed spectrum), and is applied to TDMA like channels which are shared by multiple devices. The principle of SLS was derived from the idea of *Waterfilling* [4], a well known method in the field of multiuser information theory and communications engineering.

We note that the problem of optimizing the usage of radio spectrum shared by different devices in at least one of the following dimensions: space, time, frequency, carrier, spreading code is similar to the load balancing problem in the computers [5]. The load balancing problem in computers is stated as follows: given a large number of jobs, find the allocation of jobs to computers optimizing a given objective function. In this work, we develop an algorithm for spectrum load smoothing in the time domain based on a non-cooperative game theoretic load balancing problem described in [5].

II. Spectrum load smoothing and Load Balancing

First, we outline SLS ([2], [3]) in time domain for a single fixed frequency. Also, the frame structure is fixed and SLS is applied once per frame by each requesting device. SLS allocates (time) lengths on n (time) slots to m (competing) devices over several iterations per frame (in round robin mode) as follows. Each slot has a (total) fixed slot length and the sum of its allocated lengths cannot exceed the slot length. The aggregate slots length is sufficient to accommodate all the devices' requirements. For every frame, each device considers the available slot length (remaining from previous iterations) and tries to get a portion for its requirement. The goal is to leave the slots at approximately the same load level (where load is the allocated slot length).

Remark: The time complexity of SLS algorithm is $O(n \times \sum_{i=1}^{m} iter_i)$ where n is the number of slots, m is the number of devices, and $iter_i$ is the number of iterations taken by device $i = 1, ..., m$ for SLS to converge for a single frame. For each frame, the average complexity per device is $O(n \times iter_{avg})$, where $iter_{avg} = \sum_{i=1}^{m} iter_i/m$.

In a distributed computer system consisting of n heterogeneous computers shared by m users [5], the goal of each user is to find an allocation of her/his jobs on the computers so that the average response time of her/his own jobs is minimized.

From the description of the preceding two systems, we note that the spectrum load smoothing in time domain is analogous to the load balancing in distributed system of computers [5]. The time slots of the channel in which the devices transmit their data can be thought as computers and the devices which require portions of the time slots can be thought as users. We replace 'computers' by 'time slots' (or simply 'slots') and 'users' by 'devices' in the load balancing model. We then formulate the problem for spectrum allocation as a load balancing problem in analogy to [5].

Problem Statement: Let us denote by (1) μ_i $(i = 1, ..., n)$ the length of each slot, (2) ϕ_j the length of the time required by device j $(j = 1, ..., m)$, (3) s_{ji} $(0 \leq s_{ji} \leq 1)$ the fraction (of time), such that the term $s_{ji}\phi_j$ represents the time length assigned to device j in slot i, (4) $\mathbf{s}_j = [s_{j1}, s_{j2}, ..., s_{jn}]^T$ (where T is the transpose operator) (called the load balancing strategy of device j) the fractions of time lengths of all the slots occupied by device j.

Manuscript received November 21, 2007. The associate editor coordinating the review of this letter and approving it for publication was M. Saquib. This work was supported in part by NSF grant CCR-0312323, and was presented in part at the 2^{nd} International Conference on Cognitive Radio Oriented Wireless Networks and Communications - CrownCom 2007.

A. T. Chronopoulos and M. R. Musku are with the University of Texas at San Antonio, TX 78249 (e-mail: atc@cs.utsa.edu).

S. Penmatsa is with Southern Arkansas University, Magnolia, AR 71753.

D. C. Popescu is with the Dept. of E.C.E., Old Dominion University, VA 23529.

Digital Object Identifier 10.1109/LCOMM.2008.071968.

The total time length that all the devices occupy in slot i is $\sum_{j=1}^{m} s_{ji}\phi_j$. Thus, the total amount of unused time in slot i is

$$F_i(\mathbf{s}) = \mu_i - \sum_{j=1}^{m} s_{ji}\phi_j \tag{1}$$

To achieve load balancing each device j must calculate the fraction s_{ji} so that $F_i(s)$ is maximized subject to the following constraints on (1) (analogous to [5]):

$$s_{ji} \geq 0, \quad \sum_{i=1}^{n} s_{ji} = 1, \quad \sum_{k=1}^{m} s_{ki}\phi_k < \mu_i; \quad i = 1, ..., n \tag{2}$$

The reasons are: (i) the fraction cannot be negative, (ii) for user j, the sum of fractions in all the slots should equal 1, (iii) the sum of all allocations in a slot cannot exceed the slot length.

Maximizing $F_i(\mathbf{s})$ in slot i is equivalent to minimizing $\sum_{i=1}^{n} s_{ji}/F_i(\mathbf{s})$. Therefore optimization over all slots implies that

$$D_j(\mathbf{s}) = \sum_{i=1}^{n} \frac{s_{ji}}{\mu_i - \sum_{j=1}^{m} s_{ji}\phi_j} \tag{3}$$

must be minimized. We consider this formulation to make use of the related existing game theory [5]. In particular, this allows the proof of fairness guarantees.

Therefore, here the goal of each device j can thought as selecting the strategy $\mathbf{s}$ so that the payoff function $D_j(\mathbf{s})$ is minimized. Also note that (3) is similar to the function to be optimized in load balancing problem in [5]. We formulate this problem as a non-cooperative game among devices assuming that the devices are selfish. In a non-cooperative game for spectrum allocation each device (player) with a requirement of ϕ_j determines its strategy profile $\mathbf{s}_j$ in order to minimize its corresponding payoff $D_j(\mathbf{s})$. The optimization problem associated with device j can be described as follows:

$$\min_{\mathbf{s}_j} D_j(\mathbf{s}); \quad \text{subject to the constraints in (2)} \tag{4}$$

Nash equilibrium is defined as the strategy profile for which every device's load balancing strategy is a *best reply* to other devices' strategies. This best reply for a device will find the minimum of the payoff function given the strategies of the other devices. We need to determine the strategy profile of device j which must be optimal with respect to the other devices' strategies. The solution of the optimization problem in (4) and the algorithm for calculating the optimal solution (*best reply algorithm*) and finding the Nash equilibrium was given in [5].

Best Reply Algorithm (Device-j)

Input: Available slot lengths: $\mu_1^j, \mu_2^j, ..., \mu_n^j$; Total requirement of device: ϕ_j; Output: Fractions: $s_{j1}, s_{j2}, ..., s_{jn}$

```
Initialize $k = n$
1) for $i = 1, 2, ..., n$ do
     Obtain the length of free slot available $\mu_i^j$
       $\mu_i^j \leftarrow \mu_i - \sum_{l=1, l \neq j}^{m} s_{li}\phi_l$
2) Sort the slots in decreasing order of their lengths available ($\mu_1^j \geq \mu_2^j \geq ... \geq \mu_n^j$)
3) $t \leftarrow (\sum_{i=1}^{n} \mu_i^j - \phi_j)/(\sum_{i=1}^{n} \sqrt{\mu_i^j})$
4) while ($t \geq \sqrt{\mu_k^j}$) do
     $s_{jk} \leftarrow 0$
     $k \leftarrow k - 1$
     $t \leftarrow (\sum_{i=1}^{k} \mu_i^j - \phi_j)/(\sum_{i=1}^{k} \sqrt{\mu_i^j})$
5) for $i = 1, 2, ..., k$ do
     $s_{ji} \leftarrow (\mu_i^j - t\sqrt{\mu_i^j})\frac{1}{\phi_j}$
6) Allocate in each slot $i = 1, 2, ..., n$
     $s_{ji}\phi_j$, $j = 1, 2, ..., m$
```

In order to obtain the equilibrium allocation, we need an iterative algorithm where each device updates his strategies (by computing his best response) periodically by fixing the other devices' strategies. The devices iteratively apply the *best reply* algorithm to compute the Nash equilibrium as shown below.

SLB Algorithm

Each device $j = 1, \ldots, m$ performs the following steps in each iteration:

1. Receive a message containing the current strategies of all other devices.
2. If the message indicates a termination, then broadcast the termination message and EXIT.
3. Update the strategies ($s_{ji}\phi_j$) by calling the *Best Reply* algorithm.
4. Check if the desired error norm is reached.
5. Broadcast the updated strategies and the error norm.

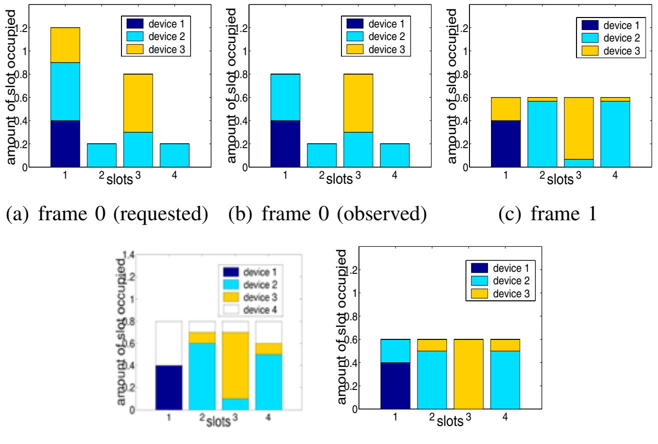
The algorithm is executed during the coordination period of each frame assuming that each device has received the allocations of all other devices before starting its allocations in the current frame as in the case of SLS. Otherwise, the less accurate allocations of the last frame are considered. The broadcast is implemented via in a base station in a cellular network or by a broadcast algorithm in ad hoc networks.

Remark: (a) The time complexity of the *best reply* algorithm is $O(n \log n)$ [5]. (b) The time complexity of SLB algorithm is $O(m \times n \log n \times iter)$ where n is the number of slots, m is the number of devices, and $iter$ is the number of iterations taken by the algorithm to converge to the Nash equilibrium. We note that $iter$ is the same for all devices unlike SLS.

III. Simulations

In this section, we compare the spectrum load balancing (SLB) introduced in the section above with the spectrum load smoothing (SLS) algorithm based on reservation [3] in terms of the achieved throughput. The normalized throughput $\Theta^j(n)$ represents the share of capacity a device j demands in frame-n, and is defined in [3]. The simulations were performed using Matlab. It is assumed that there are four devices (users) in the system and a frame structure of four time slots with a maximum load capacity of 0.8 is considered as in [3]. The remaining capacity is left unallocated to enable the additional SLB using devices or legacy devices. The maximum load level is respected by all devices and they abort their allocations if it is exceeded. The SLB is achieved over the complete frame.

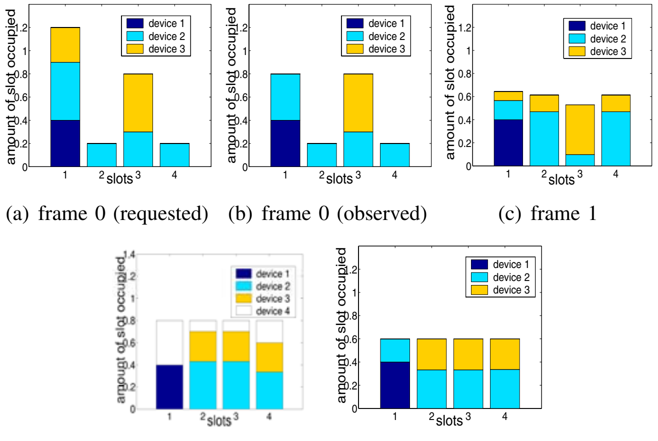
Fig. 1 shows the requested allocation (frame 0) and the allocations at frames 1, 26 and 75 of the SLS iteration. Fig. 2 shows the requested allocation (frame 0) and the allocations at frames 1, 26 and 75 of the SLB iteration. Device-j, ($j = 1, 2, 3$) share the medium during the initial frame-0 and their demanded allocations are not coordinated, i.e. they overload the first time slot leading to a shortened observed allocation for



(a) frame 0 (requested) (b) frame 0 (observed) (c) frame 1

(d) frame 26 (e) frame 75

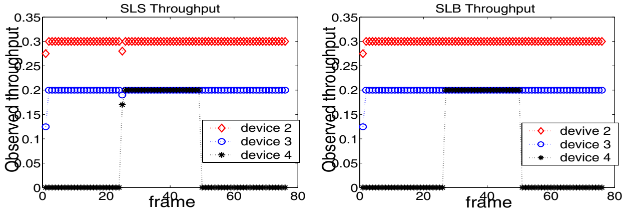
Fig. 1. SLS Observed at frames 0, 1, 26 & 75



(a) frame 0 (requested) (b) frame 0 (observed) (c) frame 1

(d) frame 26 (e) frame 75

Fig. 2. SLB Observed at frames 0, 1, 26 & 75



(a) SLS (b) SLB

Fig. 3. Observed throughput allocations of SLS and SLB

TABLE I

NUMBER OF ITERATIONS AND COMPLEXITY FOR SLS AND SLB WITH INCREASING NUMBER OF DEVICES

No. of Devices	Iterations		No. of Comps/Device	
	SLS $iter_{avg}$	SLB $iter$	SLS	SLB
4	7	2	112	64
6	10	2	240	96
9	29	2	1044	144
12	28	3	1344	288
15	33	3	1980	360

device-2 and no allocation for device-3 as depicted in Fig. 1 and Fig. 2, both in the case of SLS and SLB. During this frame the observed throughput of both the devices-j ($j = 2, 3$) is less as seen from Fig. 3. The SLB leads to mutually coordinated output of the demanded allocations during frame-1, although in a different pattern from the SLS as seen in Figs. 1 and 2.

Device-4 initiates its transmission in frame-25, demanding a share of 0.2 from the capacity. This leads to a non-coordination among the devices and a decrease in throughput of all the devices using SLS, but in SLB, since the demanded allocations is divided into parts and allocated in the same step. This effect does not occur as it is seen in Fig. 3. All the devices follow SLB and redistribute their allocations in a coordinated way and reach a steady point of interaction which is the Nash equilibrium of the game. At frame-50, device-4 terminates its transmissions, and this results in the re-distribution of allocations as seen in Fig. 1 and Fig. 2.

Table I shows the number of iterations and computations for SLS and SLB (computed using the time complexities discussed in the above remarks) with increasing number of devices. It can be observed that the devices take significantly fewer iterations in SLB compared to the SLS. Thus, the maximum number of computations required by SLB are significantly fewer than that of SLS.

IV. CONCLUSIONS

In this work, we proposed a load balancing algorithm based on game theory for spectrum allocation. It is shown that the throughput obtained by devices using SLB is same as that of SLS although the allocations of devices in the slots are different. Based on our simulations, SLB is more efficient than SLS. Another advantage of SLB is that it is based on non-cooperative game theory and it provides a (Nash equilibrium) solution which is optimal for all users. This is especially useful because the users sharing an open spectrum often act in a selfish manner.

REFERENCES

[1] S. Mangold, Z. Zhong, K. Challapali, and C.-T. Chou, “Spectrum agile radio: radio resource measurements for opportunistic spectrum usage,” in *Proc. IEEE Global Telecommunications Conference 2004 (GLOBECOM ’04)*, vol. 6, Dallas, TX, Nov. 2004, pp. 3467–3471.

[2] L. Berlemann and B. Walke, “Spectrum load smoothing for optimized spectrum utilization–rationale and algorithm,” in *Proc. 2005 IEEE Wireless Communications and Networking Conference*, vol. 2, New Orleons, LA, Mar. 2005, pp. 735–740.

[3] L. Berlemann, S. Mangold, G. R. Hiertz, and B. Walke, “Spectrum load smoothing: distributed quality-of-service support for cognitive radios in open spectrum,” *European Trans. Telecommun.*, vol. 17, pp. 395–406, Mar. 2006.

[4] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, “Iterative water-filling for Gaussian vector multiple-access channels,” *IEEE Trans. Inform. Theory*, vol. 50, no. 1, pp. 145–152, Jan. 2004.

[5] D. Grosu and A. Chronopoulos, “Noncooperative load balancing in distributed systems,” *J. Parallel and Distributed Computing*, vol. 65, no. 9, pp. 1022–1034, Sept. 2005.