A Game Theoretic Approach for Medium Access of Open Spectrum in Cognitive Radios

Madhusudhan R. Musku[1] *Student Member, IEEE*   Anthony T. Chronopoulos[2] *Senior Member, IEEE*   Satish Penmatsa[2] *Student Member, IEEE*   and   Dimitrie C. Popescu[3] *Senior Member, IEEE*

[1] Department of Electrical & Computer Engineering
[2] Department of Computer Science
The University of Texas at San Antonio
One UTSA Circle, San Antonio, TX 78249

[3] Department of Electrical & Computer Engineering
Old Dominion University
Norfolk, VA 23529

Abstract— **The requirement of additional spectrum for the ever increasing demand to support the various Quality of Service requirements (QoS) in wireless systems is of primary interest to the research community since the unlicensed spectrum is reaching its limit and regulatory changes to provide portions from licensed bands are complicated and usually take a long time. In this work, we discuss the medium access control of open spectrum for spectrum agile radios that use spectrum opportunistically, also referred to as cognitive radios. Spectrum agile radios operate in parts of the spectrum that is originally licensed to other radio services. They identify the parts of the spectrum that is unused, coordinate its usage and release it when it is required by the licensed radio system. In this work, the problem of spectrum allocation is shown to be similar to the load balancing problem in distributed computer systems, and the problem of spectrum sharing is formulated as a non-cooperative game. We propose a non-cooperative load balancing algorithm, referred here as Spectrum Load Balancing (SLB) algorithm, and is applied to spectrum agile radio system. The game has a Nash equilibrium and SLB converges to this equilibrium. In this work, the capability of SLB to support QoS in the presence of other competing cognitive networks is evaluated via simulations and compared with the Spectrum Load Smoothing (SLS) algorithm.**

I. Introduction

Wireless Communication is requiring additional spectrum to satisfy the demand for various applications and data rates. At the same time, many of these applications have increasing restrictions to spectrum access. The currently available unlicensed spectrum is reaching its limit. The licensed spectrum is limited and the new spectrum will not be available soon, as regulatory changes of the regulatory status from licensed to unlicensed bands are complicated and usually take a long time. Today, many frequency bands are often unused, for instance frequencies licensed for TV/radio broadcasts or public safety services. Support of Quality of Service (QoS) is difficult because of the missing coordination between the different radio systems operating in the same frequency band. Further more, future radio systems are required to support QoS in a shared spectrum i.e. in the presence of other radio systems. The regulation authorities therefore are re-thinking their way of spectrum licensing and the regulation of spectrum access. Flexible, dynamic spectrum usage provides a way out of the regulatory dilemma, and also the need to combine and use the free spectrum to support the current license holders. IEEE 802.22^{TM} is being standardized as a secondary radio system operating in the licensed spectrum that is originally used for TV broadcast [2]. The U.S. DARPA Next Generation Communication (XG) Program and the 6th Framework research funding Program (FP6) of the European Union are working on flexible and dynamic spectrum usage and related impacts on spectrum regulation.

Flexible and dynamic spectrum usage requires an intelligent medium access, especially in the face of QoS support. The terms *cognitive* and *smart* radios are often used in the context of intelligent spectrum usage [7]. 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 *spectrum agile radios* [6]. Spectrum agile radios are radio systems that autonomously coordinate the usage of spectrum. 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. Co-existence of such scenarios are not addressed in the existing radio systems like IEEE 802.11(e) [10]. Spectrum utilization and the coverage area can be increased, when cognitive radios organize themselves forming a meshed wireless backbone network of infrastructure links. Related work on methods for using the spectrum in an efficient way in cognitive radios can be found in [1]–[4]. A method referred to as *Spectrum Load Smoothing (SLS)* to coordinate and optimize the usage of radio spectrum is introduced in [4], and is applied to a TDMA like channels which are shared by multiple devices. The principle of SLS was derived from the idea of *Waterfilling* [11], a well known method in the field of multiuser information theory and communications engineering. The SLS is applied and evaluated in terms of QoS of spectrum agile radios in IEEE 802.11e in [2]. The improvement in the accuracy of SLS through the usage of reservations is shown in [1].

We note that the problem of optimizing the usage of radio spectrum shared by different devices in at least one

Corresponding author: atc@cs.utsa.edu

of the following dimensions: space, time, frequency, carrier, spreading code is similar to the load balancing problem in the computers [5], [9]. 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]. In Section II we outline the spectrum load smoothing problem and method as it appeared in [4] and develop a relation between this and the load balancing in distributed computer systems. In Section III we develop an algorithm based on the load balancing problem described in [5]. A simulation introduction and comparison with the existing SLS algorithm is provided in Section IV and conclusions are drawn in Section V.

II. Spectrum load smoothing in time domain and Load Balancing

Here, a periodic frame-based MAC protocol is considered as the basis for coordination and interaction as in [1]. One device, preferably the first device that initiates a transmission, introduces a slotted time frame structure as a basis for future cooperation. A slot is a time interval during which the multiple access occurs. This slotted structure can be changed by all devices, preferably by the first device, from one frame to another, but is assumed to be fixed here. The slotting can be based for instance on the system load, individual QoS requirements of supported applications or the protected allocations of an incumbent radio system. The slotted structure is regarded as mandatory and respected by all devices. In this section we will first describe the time domain model as spectrum load smoothing in [1], [4], and the load balancing model in computers [5] and then explain the relation between both of them and develop a model solving the SLS as a load balancing problem.

A. Spectrum Load Smoothing

Fig. 1 illustrates the principle of SLS in time domain for a fixed, single frequency. The frame structure is fixed and the SLS is applied once per frame by a device. The slot length is fixed and respected by all devices. The initial two steps of the iterative algorithm are shown in Fig. 1. Each device considers the added allocations of all the other devices as the origin and calculates the load level. Device-2 is the only other device and the load level of device-1 is increased step wise beginning with the lowest allocation of device-2. The step size w is given by

$$w = \frac{amount \quad of \quad allocation \quad to \quad be \quad distributed}{number \quad of \quad slots}$$

The difference between the load level and the allocations of the other device are filled with the allocations of device-1 and the amount of allocations to be distributed are subtracted from the total amount. This procedure is repeated until the amount of allocations to be distributed is zero or it satisfies a criterion for accuracy.

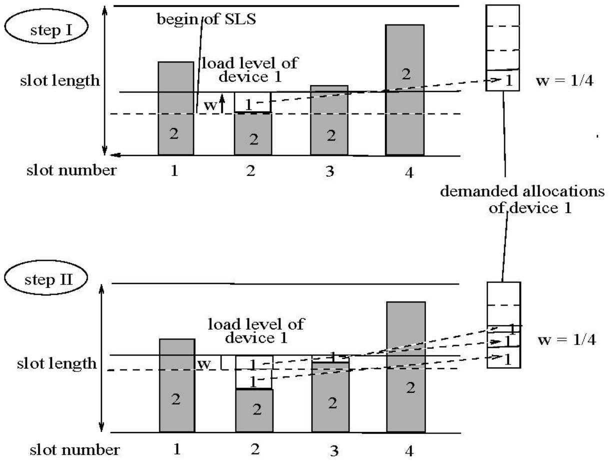


Fig. 1. Initial two steps of the principle of Spectrum Load Smoothing (SLS) in time domain

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 to reach the desired accuracy. The average complexity per device is $O(n \times iter_{avg})$, where $iter_{avg} = \frac{\sum_{i=1}^{m} iter_i}{m}$.

Fig. 2 depicts the time domain based on a slotted, periodic frame. Here, three decentralized devices coordinate each other and have periodically demanded allocations which do not necessarily have the same length, as for instance the demanded allocations of device-2. The timing diagram of the resulting channel is also shown. The first device has the most restrictive QoS requirements, by means of a single slot length as distance of smoothing. This device-1 distributes its allocations first, here under consideration of the optional coordination phase. The smoothed allocations are placed in the first slot directly after the coordination phase and in the sixth slot of the frame. The concept of SLS is observable in focusing on the allocations of the second device, device-2. With a smoothing distance of two slots under consideration of the allocations of the first device and the optional coordination phase, device-2 places most of its demanded allocation duration in the second slot and less time in the first slot. The first and second slot have equal idle duration resulting from the SLS. Device-3 initiates as the third device its transmission. After having observed at least one frame, device-3 places its allocations into the third and fourth slot of the current frame.

B. Load Balancing as a Non-cooperative game among users

A distributed computer system consisting of n heterogeneous computers shared by m users is shown in Fig. 3 [5]. The goal of each user is to find an allocation of his jobs on the computers such that the average response time of his own jobs is minimized. The average job processing rate of each computer is given by μ_i, $i = 1, ..., n$. Jobs are generated by user j with an average rate ϕ_j, and $\Phi = \sum_{j=1}^{m} \phi_j$ is the total job arrival rate in the system and $\Phi < \sum_{i=1}^{n} \mu_i$. The users have to decide on how to distribute their jobs to

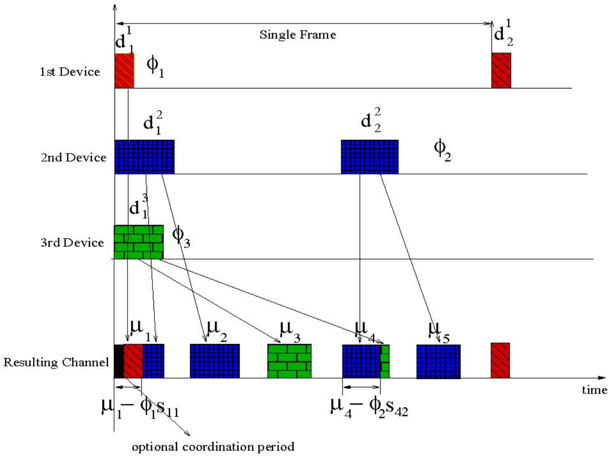


Fig. 2. SLS in time domain. The periodic allocations are smoothed.

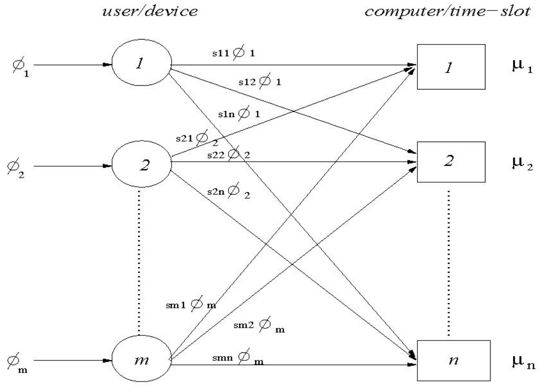


Fig. 3. Load balancing in computers/Time-slots.

computers such that they will operate optimally. Thus user j ($j = 1, ..., m$) must find the fraction s_{ji} of all its jobs that are assigned to computer i ($\sum_{i=1}^{n} s_{ji} = 1$ and $0 \leq s_{ji} \leq 1$, $i = 1, ..., n$) such that the expected execution time of its jobs ($D_j(s) = \sum_{i=1}^{n} \frac{s_{ji}}{\mu_i - \sum_{k=1}^{m} s_{ki}\phi_k}$) is minimized.

C. The Spectrum Load Smoothing expressed in terms of Load Balancing Problem Model

From the description in the above two subsections, it can be observed that the spectrum load smoothing in time domain is similar to the load balancing in computers. The fixed time slots of the channel in which the devices transmit portions of their data can be thought of computers and the devices can each be thought as a user which requires a portion of the time slot. We replace 'computers' by 'time slots' and 'users' by 'devices' in the model of Section II-B. Therefore, here in conjunction with load balancing in computers we denote the length of time slots by μ_i, and each device's requirement of the time slot with ϕ_j. Let s_{ji} denote the fraction of time that device j occupies in time slot i. Then the vector $\mathbf{s}_j = [s_{j1}, s_{j2}, ..., s_{jn}]^T$ (where T is the transpose operator) is called the load balancing strategy of device j.

The term $s_{ji}\phi_j$ represents a portion of the time that is assigned to device j in time slot i. The total amount of time that is occupied in time slot i by all the devices is $\sum_{j=1}^{m} s_{ji}\phi_j$ and the total amount of unused time in time slot i is given by

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

Each device j would like to calculate the fraction s_{ji} on slot i such that $F_i(s)$ is minimized. Then the term

$$\frac{\mu_i - \sum_{j=1}^{m} s_{ji}\phi_j}{s_{ji}} \qquad (2)$$

gives the amount of time that is available in time slot i for device j. Also,

$$\frac{s_{ji}}{\mu_i - \sum_{j=1}^{m} s_{ji}\phi_j}$$

is the inverse of the amount of available time.

The inverse of the total amount of time that is available for device j in all the time slots is

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

Therefore, here the goal of all the devices can thought of as selecting the fractions s_{ji} to each time slots such that the inverse of the total available time for each device, which hereafter is referred to as 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].

III. Spectrum Load Smoothing as a Non-cooperative game

The spectrum load smoothing problem can be stated as follows: given a m number of devices with requirements of ϕ_j and number of fixed length time slots μ_i (here it is assumed that $\mu_1 = \mu_2 = ... = \mu_n$), each device should calculate the fraction s_{ji} and transmit within time interval = $s_{ji}\phi_j$ in time slot i such that the payoff function is minimized. 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 $D_j(\mathbf{s})$.

Nash equilibrium is defined as the strategy profile for which every device's load balancing strategy is a *best reply* to other devices' strategies [8]. 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. Let $\mu_i^j = \mu_i - \sum_{k=1,k\neq j}^{m} s_{ki}\phi_k$ be the amount of slot available for device j in slot i. The problem of computing the best reply strategy of device j ($j = 1, 2, ..., m$) reduces to computing the optimal strategy for the system with one device and n slots with lengths μ_i^j and each device's requirement ϕ_j. The optimization problem associated with device j can be described as follows:

$$\min_{\mathbf{s}_j} D_j(\mathbf{s}) \qquad (4)$$

subject to the constraints

$$s_{ji} \geq 0, \quad i = 1, 2, ..., n \tag{5}$$

$$\sum_{i=1}^{n} s_{ji} = 1 \tag{6}$$

$$\sum_{i=1}^{m} s_{ki}\phi_k < \mu_i, \quad i = 1, 2, ..., n \tag{7}$$

The amount of fraction of device j in any slot cannot be negative (5) and the sum of fractions in all the slots should be equal to 1 (6). Also, the amount of allocations in a slot cannot be more than the slot length (7), which means that the slot cannot be overloaded. It can be seen from (3), that $\frac{\partial D_j(\mathbf{S})}{\partial s_{ji}} \geq 0$ and $\frac{\partial^2 D_j(\mathbf{S})}{\partial s_{ji}^2} \geq 0$ for $i = 1, 2, ..., n$; which means that the Hessian of $D_j(\mathbf{s})$ is a convex function of the strategies $\mathbf{s}_j$.

The solution to the optimization problem in (4) is given in [5] and the algorithm for calculating the optimal solution (best reply) and finding the Nash equilibrium was also developed. Here, we formally state the non-cooperative load balancing algorithm in relation with the spectrum allocation.

A. Spectrum Load Balancing

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: $s_{j1}, s_{j2}, ..., s_{jn}$

During the coordination period of each frame, device j, $(j = 1, 2, .., m)$ executes

```
1) for $i = 1, 2, ..., n$ do
     Obtain the length of free slot available $\mu_i^j$
       $\mu_i^j \leftarrow \mu_i - \sum_{k=1,k\neq j}^{m} s_{ki}\phi_k$
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 $(t \geq \sqrt{\mu_n^j})$ do
       $s_{jn} \leftarrow 0$
       $n \leftarrow n - 1$
       $t \leftarrow (\sum_{i=1}^{n} \mu_i^j - \phi_j)/(\sum_{i=1}^{n} \sqrt{\mu_i^j})$
5) for $i = 1, 2, ..., n$ do
       $s_{ji} \leftarrow (\mu_i^j - t\sqrt{\mu_i^j})\frac{1}{\phi_j}$
6) Allocate in each slot
       $s_{ji}\phi_j$
```

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. A virtual ring topology of the devices can be used to communicate and iteratively apply the *best reply* algorithm to compute the Nash equilibrium as shown below.

SLB Algorithm

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

1. Receive the current strategies of all the other devices from the left neighbor.
2. If the message is a termination message, then pass the termination message to the right neighbor 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. Send the updated strategies and the error norm to the right neighbor.

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.

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.

IV. Simulations

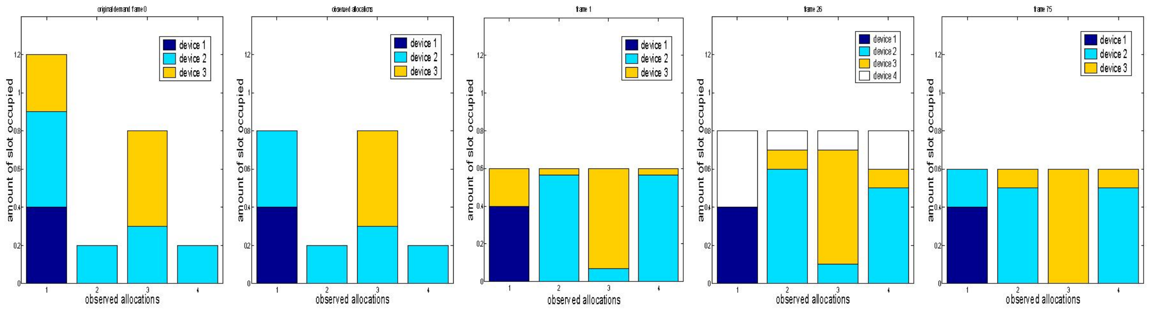
In this section, we compare the spectrum load balancing (SLB) introduced in the section above with the spectrum load smoothing (SLS) algorithm based on reservation presented in [4]. We will compare both the algorithms in terms of the achieved throughput. The normalized throughput $\Theta^j(n)$ represents the share of capacity a device j demands in frame-n, and is defined as

$$\Theta^j(n) = \frac{1}{FrameLength} \sum_{l=1}^{L^j(n)} d_l^j(n) \quad \in [0, 1] \tag{8}$$

where $L^j(n)$ is the number of allocations per frame-n, $FrameLength$ is the duration of the frame and $d_l^j(n)$ are the demands as shown in Fig. 2. 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 [4]. 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.

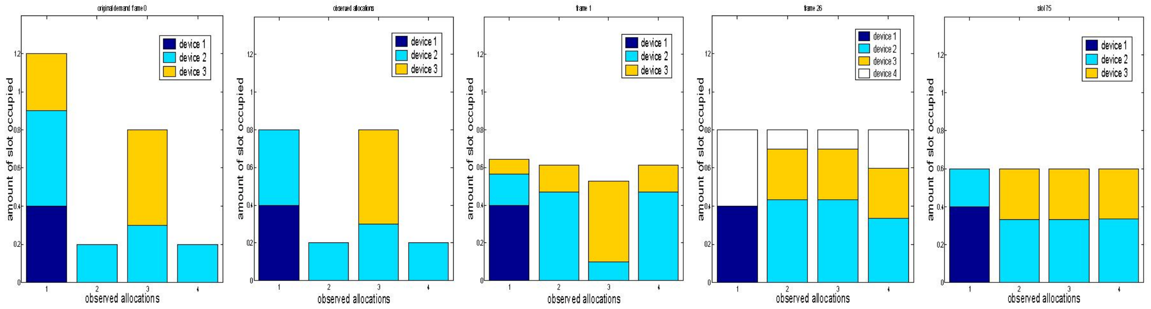
Device-j, $(j = 1, 2, 3)$ share the medium during the initial frame-0 and their demanded allocations are not coordinated, i.e. they overload the first time slot leading to a shortened observed allocation for device-2 and no allocation for device-3 as depicted in Fig. 4 and Fig. 5, both in the case of SLS and SLB. During this frame the observed throughput of both the devices-j $(j = 2, 3)$ is less as seen from Fig. 6 and Fig. 7. The SLB leads to mutually coordinated output of the demanded allocations during frame-1, although in a different pattern form the SLS as seen in Fig. 4 and Fig. 5.

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



(a) frame 0 (demanded) (b) frame 0 (observed) (c) frame 1 (d) frame 26 (e) frame 75

Fig. 4. SLS Observed at frames 0, 1, 26 & 75



(a) frame 0 (demanded) (b) frame 0 (observed) (c) frame 1 (d) frame 26 (e) frame 75

Fig. 5. SLB Observed at frames 0, 1, 26 & 75

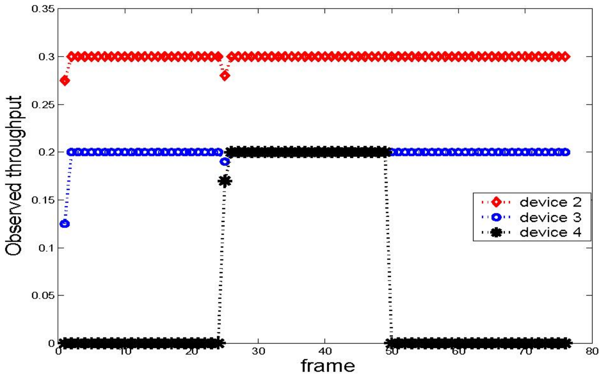


Fig. 6. Observed throughput allocations during SLS

devices using SLS, but in SLB, since the demanded allocations is divided in to parts and allocated in the same step, this effect does not occur as it is seen in Fig. 6 and Fig. 7. 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. 4 and Fig. 5.

Table I shows the number of iterations and number of computations (computed using the time complexities discussed in the above remarks) taken by SLS and SLB 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.

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

V. CONCLUSIONS

In this work, we proposed a load balancing algorithm based on game theory for spectrum allocation in a TDMA like system which is shared by multiple devices and compared it with the existing SLS algorithm. The spectrum load balancing algorithm is simple and accurate. 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. The SLB is less complex and requires fewer iterations to reach the equilibrium which makes it fast and efficient.

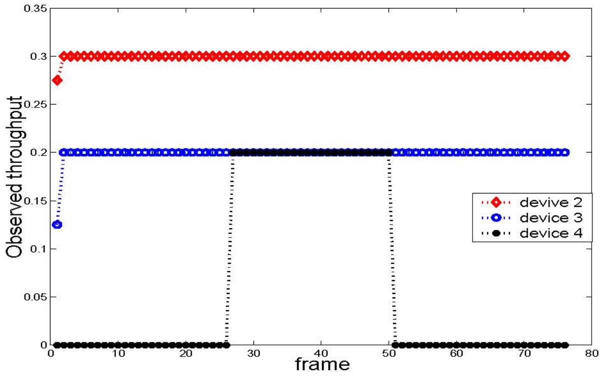


Fig. 7. Observed throughput allocations during SLB

Acknowledgment

This work was supported in part by National Science Foundation under grant CCR-0312323.

References

[1] L. Berlemann, G.R. Heirtz, and B.H. Walke. Reservation-based Spectrum Load Smoothing as Cognitive Medium Access for Spectrum Sharing Wireless Networks. In *Proceedings of European Wireless Conference 2005*, volume 2, pages 547–553, Nicosia, Cyprus, April 2005.

[2] L. Berlemann, S. Mangold, G.R. Heirtz, and B.H. Walke. Cognitive Medium Access with Spectrum Load Smoothing for Open Spectrum. In *IEEE Global Telecommunications Conference, 2005. GLOBECOM '05*, St. Louis, MO, November 2005.

[3] L. Berlemann, S. Mangold, and B.H. Walke. Policy-based reasoning for spectrum sharing in radio networks. In *First IEEE New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005*, pages 1 – 10, Baltimore, ML, November 2005.

[4] L. Berlemann and B. Walke. Spectrum load smoothing for optimized spectrum utilization - rationale and algorithm . In *2005 IEEE Wireless Communications and Networking Conference*, volume 2, pages 735 – 740, New Orleans, LA, March 2005.

[5] D. Grosu and A.T. Chronopoulos. Noncooperative Load Balancing in Distributed Systems. *Journal on Parallel and Distributed Computing*, 65(9):1022 – 1034, September 2005.

[6] S. Mangold, Zhun Zhong, K. Challapali, and Chun-Ting Chou. Spectrum agile radio: radio resource measurements for opportunistic spectrum usage . In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04*, volume 6, pages 3467 – 3471, Dallas, TX, November 2004.

[7] J. Mitola and G.Q Maguire. Cognitive Radio: Making Software Radios More Personal. *IEEE Personal Communications Magazine*, 6:13 – 18, August 1999.

[8] M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, New York, NY, 2004.

[9] S. Penmatsa and A.T. Chronopoulos. Price-based User-optimal Job Allocation Scheme for Grid Systems. In *Proceedings* 20th *IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006)*, Rhodes Island, Greece, April 2006.

[10] IEEE 802.11 WG. Draft Amendment to Standard for Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - PART II: Wireless Medium Access Control (MAC) and Physical (PHY) layer specifications: Medium Access Control (MAC) Quality of Service (QoS), IEEE 802.11e/D8.0. February 2004.

[11] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi. Iterative Water-Filling for Gaussian Vector Multiple-Access Channels. *IEEE Transactions on Information Theory*, 50(1):145 – 152, January 2004.