한국ITS학회논문지
제8권, 제3호 (2009년 6월)
pp. 71~82

Dynamic Spectrum Load Balancing for Cognitive Radio in Frequency Domain and Time Domain

Juan Chen* 손 성 환** Junrong Gu** 김 재 명***
(Juan Chen) (Sung-Hwan Sohn) (Junrong Gu) (Jae-Moung Kim)

Abstract

As a solution to spectrum under-utilization problem, Cognitive radio (CR) introduces a dynamic spectrum access technology. In the area, one of the most important problems is how secondary users (SUs) should choose between the available channels, which means how to achieve load balancing between channels. We consider spectrum load balancing problem for CR system in frequency domain and especially in time domain. Our objective is to balance the load among the channels and balance the occupied time length of slots for a fixed channel dynamically in order to obtain a user-optimal solution. In frequency domain, we refer to Dynamic Noncooperative Scheme with Communication (DNCOOPC) used in distributed system and a distributed Dynamic Spectrum Load Balancing algorithm (DSLB) is formed based on DNCOOPC. In time domain, Spectrum Load Balancing method with QoS support is proposed based on Dynamic Feed Back theory and Hash Table (SLBDH). The performance of DSLB and SLBDH are evaluated. In frequency domain, DSLB is more efficient compared with existing Compare_And_Balance (CAB) algorithm and gets more throughput compared with Spectrum Load Balancing (SLB) algorithm. Also, DSLB is a fair scheme for all devices. In time domain, SLBDH is an efficient and precise solution compared with Spectrum Load Smoothing (SLS) method.

Key words: Cognitive radio, opportunistic spectrum access, dynamic spectrum load balancing, non-cooperative game theory, hash table, feed back theory

† 이 논문은 2006년도 정부(과학기술부)의 재원으로 한국과학재단의 국가지정연구실사업으로 수행된 연구임. (No. M10600000194-06J0000-19410).
† 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음. (IITA-2009-C1090-0902-0019).
* 주저자 : 인하대학교 정보통신대학원 WiTLAB 석사과정(교신저자)
** 공저자 : 인하대학교 정보통신대학원 WiTLAB 박사과정
*** 공저자 : 인하대학교 정보통신대학원 WiTLAB 교수
† 논문접수일 : 2009년 5월 20일
† 논문심사일 : 2009년 6월 22일
† 게재확정일 : 2009년 6월 23일

I. Introduction

Most of the radio spectrum relevant to wireless communications is densely allocated by regulatory. It shows inefficient use in licensed spectrum, where appears some "spectrum holes". In addition, operators in unlicensed frequency bands engender much heavier spectral usage [1]. Under this situation, Cognitive Radio (CR) is emerging as an opportunity to enable dynamic and real-time opportunistic spectrum access and originally introduced by Mitola [2]. CR is an intelligent wireless communication system that is aware of its surrounding environment and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimulation by making corresponding changes in certain operating parameters [3]. CR devices can sense spectrum holes when they are unused by the incumbent radio system and access into spectrum hole without interference with other devices (users).

Actually, there are two problems concerning to opportunistic spectrum access. One is that the secondary users (SUs) should find out the spectrum holes and decide how to use them. The other one is how SUs should choose between the available channels, which means how to achieve load balancing between channels in frequency domain [4] or how SUs achieve load balancing in time domain for a fixed channel. Many efforts have been made in load balancing field. First, in frequency domain, CAB algorithm is proposed in [4] to achieve an equilibrium solution. Because the spectrum access is based on probability in [4], there may be redundantly repeated times to get load balancing. The efficiency of this algorithm should be queried. Another problem is that all SUs are assumed to impose the same traffic load to the system. So this algorithm cannot make a general application due to the complicated real world conditions. Second, in time domain, water filling method is introduced in [5] and game theory is used in [6]. In [5], the application of "waterfilling" from the information theory, referred to as Spectrum Load Smoothing (SLS) is proposed. In time domain for a single fixed channel, SLS is applied once per frame by each requesting device and the frame structure is fixed. SLS can precisely carry out load balancing, however, SLS costs too much iteration to converge. SLB algorithm is proposed in [6]. However SLB is a static solution for load balancing, so it is not feasible to real-time systems.

In this paper, we consider load balancing problem both in frequency and time domain. First, we get a solution of Dynamic Spectrum Load Balancing (DSLB) for frequency domain based on [7] and [8]. It is assumed that each device may have different load, which makes our work more practical. Our objective is to balance the load (the amount of processing rate) among the channels dynamically in order to obtain a user-optimal solution (i.e. to minimize the expected response time or to maximize the throughput of the individual users). In this paper, we only consider the systems of which channel capacity is elastic such as Code Division Multiple Access (CDMA). Devices access and are handed over dynamically, so we must dynamically balance spectrum load on the basis of changing channel capacity and temporary devices. This is similar to dynamic multi-user load balancing in distributed systems [7]. Second, our focus is efficient load balancing method in time domain. In time domain for a fixed channel, we propose the Spectrum Load Balancing supporting QoS based on Dynamic Feed Back and Hash table (SLBDH). Here the load is the amount of time length occupied at each slot per frame.

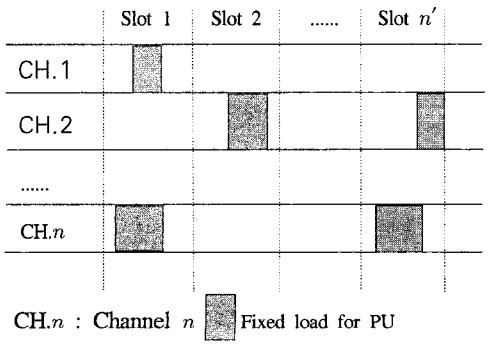
To validate the analytical results, simulations have been conducted. For frequency domain, DSLB comes close to a balancing state more quickly than CAB method and can achieve better throughput than SLB method. Also, DSLB is a fair scheme for all devices. For time domain, SLBDH method is an efficient and

precise solution compared with SLS method. The effect of intersected granularity to time complexity is also checked.

The paper is organized as follows. In Section Ⅱ we describe our system model. DSLB and SLBDH algorithms are given in Section Ⅲ.Our simulation results are presented in Section Ⅳ. Finally, concluding remarks are made in Section Ⅴ.

Ⅱ. System Model

The following system model is considered. We are given m devices (users) with a cognitive radio capability and n channels. Each channel i (i=1,⋯,n) has an elastic capacity bound μ_i. For each single channel, the slotted structure should be respected by all devices. The frame consists of n' slots with equal durations and it is should allocated n' time length on slots to m' ($0 < m' \leq m$) devices accessing into local channel over several iterations per frame. Access model is shown in <Fig. 1>. All devices need to know the slot structure prior or have to learn it from spectrum sensing [5]. In a distributed environment, the slot length can be identified with the help of the autocorrelation function of the observed allocation [9]. The coming events of devices are all assumed to follow Poisson process and their channel occupying time follow negative exponentially distributed.



<Fig. 1> Access model

We use the notation in [7] for reference and also introduce some additional notations as shown in appendix Ⅰ.

Ⅲ. DSLB and SLBDH algorithms

DSLB is used for each SU deciding to access into proper channel and handed over if necessary. If SU accesses into one channel, SLBDH is carried out on time domain per frame.

1. DSLB algorithm

SU should decide whether it needs to stay in the local channel or changes to another channel to achieve the best possible performance [4]. Because each device has a Poissonian living-time, new devices can access to the channel and leave the channel at different time which means there will be fluctuations of events. The accessing performance of devices can be regarded as dynamic non-cooperative game between devices, so it is almost the same as DNCOOPC in [7]. DSLB balances the load among the channels dynamically in order to obtain a user-optimal solution (i.e. to minimize the expected response time or maximize the throughput of the individual devices). Here it is assumed that throughput is the reciprocal of expected response time.

The expected response time of a device j (j=1,⋯,m) processed at channel i is denoted by $F_i^j(\lambda_i)$.

The expected communication delay of a device j from channel r to s is denoted by $G^j(\beta)$.

The expected response time of device j (j=1,...,m) is :

$$D^j(\lambda) = \frac{1}{\phi^j}\left[\sum_{i=1}^{n}\lambda_i^j F_i^j(\lambda_i) + \beta^j G^j(\beta)\right] = \frac{1}{\phi^j}\left[\sum_{i=1}^{n}\frac{\lambda_i^j}{\mu_i - \sum_{k=1}^{m}\lambda_i^k} + \frac{\beta^j t}{1 - t\sum_{k=1}^{m}}\beta^k\right] \quad (1)$$

The goal of each device can be thought as selecting the proper strategy λ^j, so that the payoff function $D^j(\lambda)$

is minimized. The optimization problem associated with device j can be described as follows:

$$\min_{\lambda^j} D^j(\lambda) \tag{2}$$

$$\lambda_i^j \geq 0, \sum_{i=1}^{n} \lambda_i^j = \phi^j, \sum_{j=1}^{m} \lambda_i^j < \mu_i, i = 1,\ldots,n \tag{3}$$

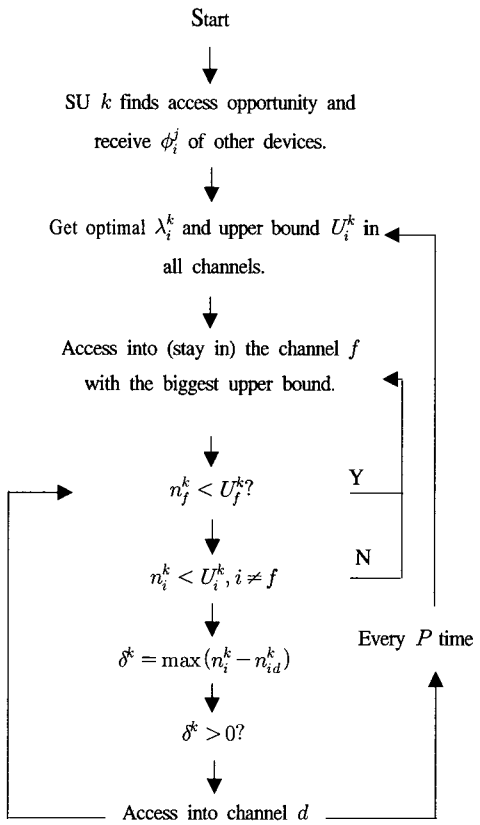
The concept of Nash equilibrium is a solution of this non-cooperative game [10]. At Nash equilibrium, each device's strategy is the best reply to other devices' strategies and each device would not decrease their own expected response time while other device's strategies are fixed. [8] offers a optimal algorithm of solution (Best Reply Algorithm).

[7] uses the number of events n_i^j (i=1,⋯,n; j=1,⋯, m) of each device in the channel as the state information to be exchanged between the devices.

$$n_i^j = \frac{\lambda_i^j}{\mu_i - \sum_{k=1}^{m} \lambda_i^k}, \tag{4}$$

The DSLB algorithm flow chart is shown in <Fig. 2>. Firstly, SU finds the optimal strategy λ_i^k (i=1,⋯,n,k= 1,⋯,m) and decides the upper bound of load U_i^k.

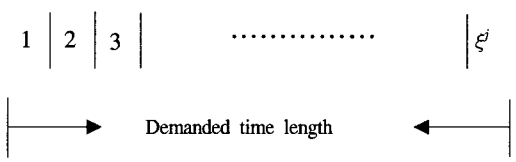
Then, SU accesses into the channel f with the biggest upper bound. Once loads exceed the upper bound, SU according to Proposition 4.2 in [7] finds another channel d with the lightest loads. Every P time, SU recalculates the optimal strategy based on the practical condition of system and reconsiders the channel with the biggest upper bound.



<Fig. 2> DSLB algorithm flow chart

2. SLBDH algorithm

For a fixed channel, each time cell of the demanded length is used for allocating. First, each demanded time length w^j (j=1,⋯, m') of each device j is cut into short time cells and the intersected granularity, denoted by l, are equal for all devices. Intersected granularity means the length of time cell. The number of time cells is denoted by ξ^j. <Fig. 3> illustrates the division of demanded time length. Then time cell p (p=1,⋯,ξ^j) is determined which slot may be occupied. Because each device has to deal with a large amount of data and reasonable data structure will have a Multiplier Effect, for each device Hash Table is used for mapping the hash key number K_p^j which represents time cell p of device j (p=1,…, ξ^j; j=1,…,m) to slot number S_p^j. In addition, Dynamic Feed Back method is used to get the allocation of load.



<Fig. 3> Division of demanded time length

<Table 1> Hash table

K_1^j	S_1^j
K_2^j	S_2^j
K_p^j	S_p^j
...	...
$K_{\lambda^j}^j$	$S_{\lambda^j}^j$

1) Hash table

Hash table is the mapping of hash key number K_p^j to slot number S_p^j, shown in table 1. Hash key number K_p^j can be got with the following expression.

$$K_p^j = H^j \oplus L^j \oplus N \tag{5}$$

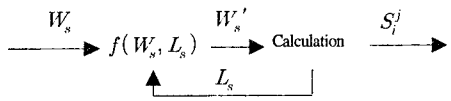
Here H^j denotes High 16 bits of binary ID of SU j, L^j means Low 16 bits of binary ID of SU j and binary N represents the "N th" time cell to be assigned. Here ID can be the low 32 bits of MAC address or the low 32 bits of international mobile subscriber identity (IMSI) or Electronic Serial Number (ESN).

Each device j calculates the hash key number K_p^j (p =1,$\cdots$,ξ^j). If this hash key number K_p^j is mapped to slot number S_p^j in hash table, the desired record would be acquired directly. Otherwise device j invokes Dynamic Feed Back method with slot weight to get slot number.

2) Dynamic Feed Back Method with Slot Weight (DFBSW)

We will outline the theory of Dynamic Feed Back with Slot Weight (DFBSW) here. DFBSW dynamically adjusts the demanded proportions of each slot to avoid overloading. Weight of each slot is used to reflect load of slot.

<Fig. 4> illustrates the structure of DFBSW. It is assumed that from $f(W_s, L_s)$ we can compute new weight W_s' of each slot based on aggregate load $L_s$and weight W of each present slot. If the difference l between new load L_s' and old load L_s exceeds a



<Fig. 4> Dynamic Feed Back theory model

threshold value of t_a, W_s' will replace W. Otherwise W remains unchanged.

$$W_s' = \begin{cases} W_s - l & (l > t_a) \\ W_s & (l \le t_a) \end{cases} \tag{6}$$

Definition 3.1:

$W_s = \mu_s' - L_s; L_s = L_{fs} + L_{ds}$; $L_{ds} = L_{ds} + l$.

Once there is a new time cell assigned to slot s, the dynamic load of slot L_{ds} will be advanced by l. The initial value of L_{ds} is 0. In CR system, primary users (PU) may have fixed frame structure and occupy the slot with fixed load. As far as devices concern, the allocated load of other SU devices is considered as fixed load. So for SU j aggregate load of each slot L_s should include two parts, one is fixed load of PU and load occupied by other SU, denoted by L_{fs}, the other one is dynamic load L_{ds} of SU j. We denote μ_s' as the length of each slot. It is assumed that there should be some time length left at each slot for the additional SU or PU. So μ_s' is set to be 80% of the capacity of each slot.

Definition 3.2:

if $\frac{L_d}{W_d} = \min\left(\frac{L_s}{W_s}\right)$,$s$=(1,...,$n'$,$d \in s$), then time cell is assigned to slot d.

Time cell is always preferred to be allocated into the slot having maximal ratio of load to weight.

3) QoS support

Each SU is assumed to detect spectrum holes and accesses into the proper hole with proposed DSLB method. Then, SU performs SLBDH for fixed channel.

The decentralized QoS support in operation at shared frequencies is one of the key challenges for future wireless communication [5]. For the sake of QoS, We assume there is a queue for all SUs and each SU takes turns to perform SLBDH algorithm according to its location in the queue. The queue is sorted in descending order basing the value of loss time, for the same loss time, in accordance with the priority ranking from large to small. Loss time represents the maximum endurable service time.

4) SLBDH algorithm with QoS support

We outline SLBDH algorithm supporting QoS as follows and <Fig. 5> shows the flow chart of SLBDH algorithm:

After accessing into channel, SUs decide the time load allocations at the beginning of a frame.

1) From the previous frame, SU knows the loss time and priority of all the other devices. Then SU gets to know the location of itself in the queue. If new devices intend to access into the current frame, the hash table of each SUs will be clear up.

2) In order, SU decides the demanded allocations of communications according to the queue at the beginning of each frame.

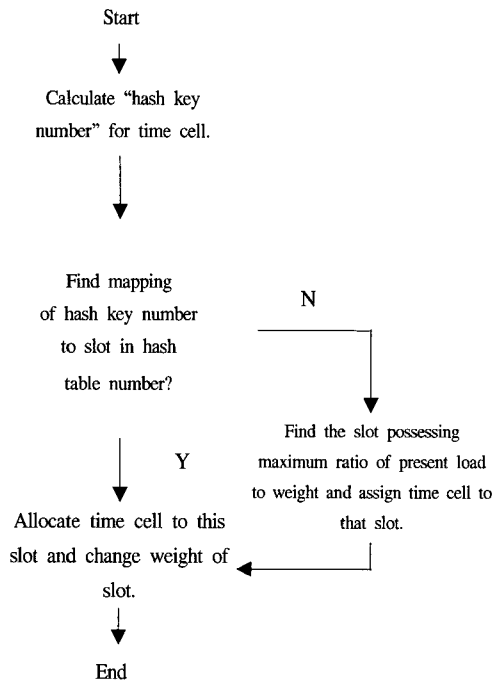
a) SU divides the demanded load into time cells as mentioned before.

b) Use (5) to get hash key numbers for each time cell.

c) Look up hash table to find the hash key numbers and the corresponding records of slots.

i) For each time cell, if there is the record of the hash key number and the relevant slot number. SU assigns the time cell to the slot; if not, jump to the next step.

ii) For each time cell, if there is no record of the hash key number, SUs perform DFBSW method mentioned before. Then the hash key number and the chosen slot number will be added to hash table.



<Fig. 5> SLBDH algorithm flow chart

Ⅳ. SIMULATION

To validate the analytical results, simulations are conducted. For frequency domain, DSLB is compared with CAB on converging to load balancing. Comparing SLB, throughput is also checked. As to the time domain mathod, SLBDH method is compared with SLS in terms of efficiency and throughput.

1. Simulation for DSLB algorithm

We will evaluate the performance considering a network by 10 channels and 100 devices (users). For the sake of simplicity, we assume there are 4 channels respectively having channel capacity as 10, 3 channels respectively having channel capacity as 20, 2 channels respectively having channel capacity as 50 and the other one has channel capacity as 100. The access rate for each device ϕ^j (j=1,…, 100) is determined from ϕ^j

<Table 2> Channel capacity of 10 channels

No.	1~4	5~7	8, 9	10
Channel capacity	10	20	50	100

<Table 3> Device access rate fraction S^j

Device	1~20	21~30	31~60	61~70	71~100
S^j	0.012	0.03	0.015	0.01	0.007

$=\Phi s^j$. s^j are given in <Table 3>. The mean communication time t for sending or receiving a device from one channel to another for any device is set to be 0.01 sec. Threshold time P is set to be 1 sec.

1) Convergence

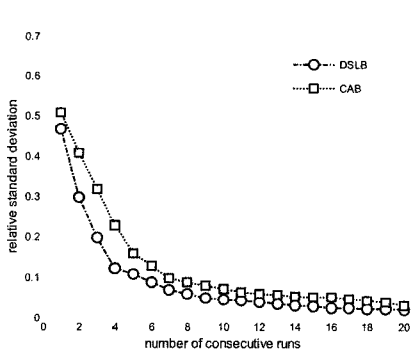
When each device reaches Nash Equilibrium, the expected response time of each device $D^j(\lambda)$ should be minimum so the $f_i^j(\lambda_i)$ of each device at all the channels are balanced and the load of each channel $\lambda=[\lambda_1,\lambda_2,...,\lambda_n]^T$should be most equal. We use the relative standard deviation of load as a measure of the balance of the system. Relative standard deviation is the ratio of standard deviation to arithmetic mean. We use relative standard deviation to normalize the maximal standard deviation to be 1. The arithmetic mean of load when the system reaches Nash Equilibrium is the base value of standard deviation which is used to measure the degree deviating from the base value. At random, total device access rate of the system Φ is assumed set to be 180.

<Fig. 6> shows how the relative standard deviation of channel load decreases over time. DSLB comes close to Nash Equilibrium more quickly than CAB method. After about 7 consecutive runs, DSLB almost gets to Nash Equilibrium and the relative standard deviation of channel load decreases to below 3%.

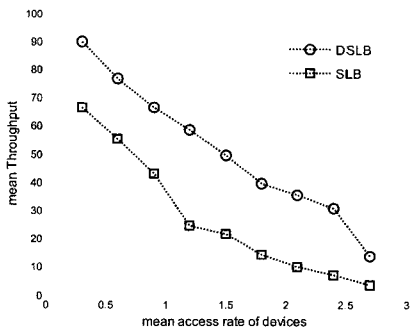
2) Throughput

Throughput means the completed amount of communication per second. Here we use the reciprocal of the expected response time as throughput. As mean access rate of device increases, mean throughput of system decreases, which we can see clearly from <Fig. 7>. All the devices share the spectrum so inevitably the throughput of system would be decrease with the system utilization increases. DSLB can achieve better throughput based on real-time changes than SLB method in [6] which only pays attention to the previous state.

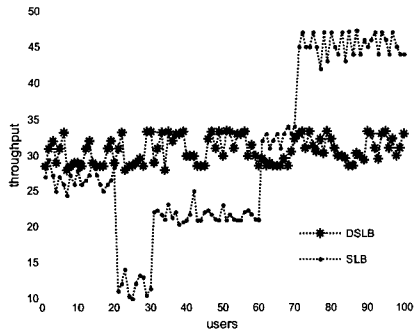
We assume that mean access rate of device is 180. In <Fig. 8>, it can be observed that throughput of all the devices are almost around 30 for DSLB algorithm



<Fig. 6> Relative standard deviation of the channel load for the algorithms COMPARE_AND_BALANCE (CAB) and DSLB



<Fig. 7> System mean throughput with mean access rate of device increase



<Fig. 8> Expected throughput of each device for algorithms DSLB and SLB

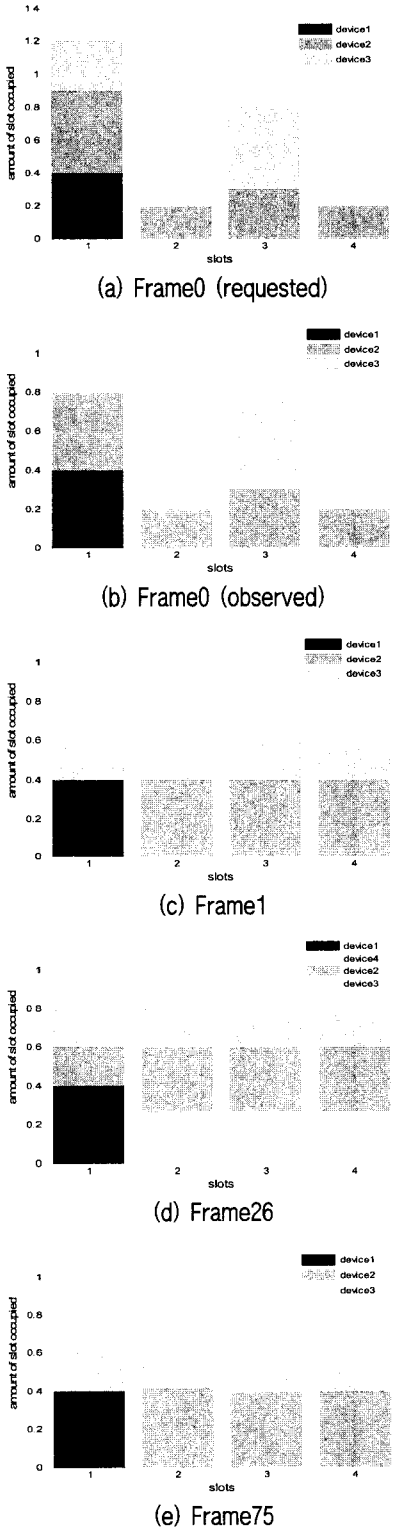
otherwise in the case of SLB there is large differences in the users' expected throughputs. DSLB provides almost equal throughput for all devices.

2. Simulation for SLBDH

For the simplicity, parameter setting is the same as [5]. Each frame is consisted of four slots with the normalized maximum slot load level of 0.8. It is assumed that there are three SU in the system and PU has fixed normalized time load of 0.4 at slot 1. So the respectively initial normalized load of each slot is 0.4, 0, 0, and 0. The relevant weight of each slot is 0.4, 0.8, 0.8 and 0.8. The maximum load level is respected by all devices and they abort their allocations if it is exceeded. We assume that device 1 is PU device, device 4 device 2, device 3 and are SU devices and they operate SLBDH algorithm in turn. It is assumed the Intersected granularity l for each device is 1/10000 and the value of t_a is 0.

1) Observed load balancing

<Fig. 9> shows the results of spectrum allocation of four devices from frame 1 to frame 75. <Fig. 9(a)> and <Fig. 9(b)> show all devices do not coordinate with each other and they know the demanded load of other devices. Device 2 and device 3 have already exceeded



(a) Frame0 (requested)

(b) Frame0 (observed)

(c) Frame1

(d) Frame26

(e) Frame75

<Fig. 9> SLBDH observed at frame 0, 1, 26, and 75

the capacity of slot 1 at frame 0, so they cannot reach their demanded load in slot 1. At the beginning of frame1, each device operates SLBDH algorithm and each slot achieve load balancing. It is assumed device 4 accesses into the frequency from frame 26 and device 4 has the highest priority so it is first to allocate time length. After frame 74, device 4 is assumed departing from the frequency. Then three devices left still comply with the order in their queue to perform spectrum allocation.

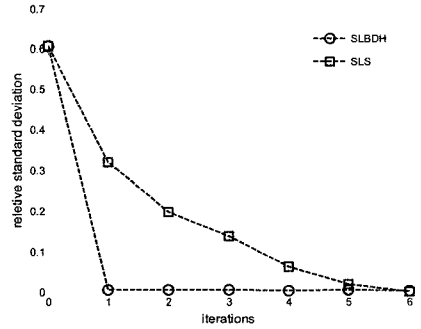
2) Convergence

Our SLBDH algorithm compares with SLS algorithm in terms of the relative standard deviation of load. Standard deviation is used to measure the differentiation between the real value and standard value. Relative standard deviation is the normalized standard deviation and is the ratio of standard deviation to standard value. Here we assume the theoretical value of balanced load of slots is the standard deviation. The smaller the value of relative standard deviation is the closer to the balanced load of slots.

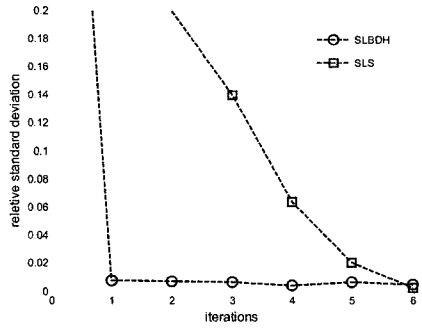
<Fig. 10(b)> is zoomed <Fig. 10(a)>. As shown in <Fig. 10>, Obviously SLBDH can achieve balancing state more quickly than SLS. After once iteration, SLBDH almost already approaches to load balancing which SLS achieves after 6 iterations.

3) Throughput

<Fig. 11> shows the throughput of algorithms SLBDH and SLS. The normalized throughput $\Theta^j(n)$ means the share of capacity a device j demands in frame-n and is defined in [10]. Device 4 initiates its transmission at frame 25. Load balancing at frame 26 is

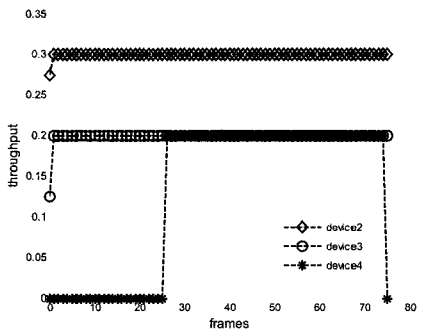


(a) Relative standard deviation of load values

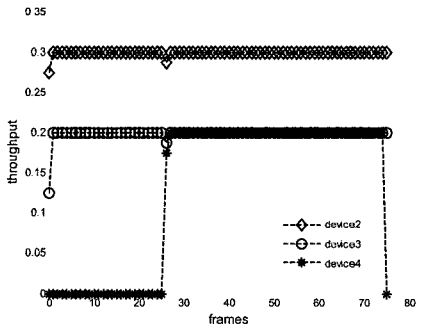


(b) Zoomed of (a)

<Fig. 10> Relative standard deviation of load values for the algorithms SLBDH and SLS



(a) Throughput of SLBDH



(b) Throughput of SLS

<Fig.11>Throughput for algorithms SLBDH and SLS

shown in <Fig. 9(d)>.

There is non-coordinated at frame 25 in SLS. However the SLBDH throughput of device 4 at frame 25 is not decreased. This because devices are sorted in descending order before they perform SLBDH algorithm and the demanded allocations are divided into cells. At frame 75, device 4 departs from the frequency; the left devices clear up their hash table and redistribute time length as show in <Fig. 9(e)>.

4) The Relation between Granularity and Time Complexity

The maximal time complexity of SLBDH algorithm is $O\left(m' \times \frac{\max(w^j)}{l} \times (n'+1)\right)$, $(j=1,\cdots, m')$. The average he average complexity per device is $O\left(\frac{\max(w^j)}{l} \times (n'+1)\right)$. We can see clearly that granularity could affect the complexity. Obviously, if the intersected granularity is smaller, the SLBDH algorithm can be more precise. However, the time complexity would also becomes bigger. The proper balance between intersected granularity and time complexity can be found according to the application requirement.

V. Conclusions

In this paper, we propose two methods to deal with load balancing problem in CR system both in frequency domain and time domain. Especially, we focus more on spectrum load balancing problem in time domain.

In frequency domain, it is formed a distributed Dynamic Spectrum Load Balancing algorithm (DSLB) with elastic channel capacity referring to [7] and [8]. We compare DSLB with Compare_And_Balance (CAB) method for efficiency and Spectrum Load Balancing (SLB) supporting QoS based on Dynamic Feed Back theory and Hash Table (SLBDH). SLBDH is compared with Spectrum Load Smoothing (SLS) in terms of efficiency and throughput. The relation between granularity and time complexity is also studied. SLBDH is more efficient than SLS and the throughput are almost the same except some particular frames. The smaller the intersected granularity is the more precise SLBDH algorithm can be but the time complexity is bigger. So it must be found proper balance between intersected granularity and time complexity for practical applications. Another advantage of SLBDH is that it can provide QoS for devices with the sorting of devices.

However, the number of calculation per device is heavy, so DSLB and SLBDH can be used at appropriate occasions or modified in the future.

References

[1] C. Cordeiro, K. Challapali, D. Birru, and S. Sai, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," *Proc. IEEE Intern. Symp. New Frontiers in Dynamic Spectrum Access Networks*, pp. 328-337, Nov. 2005.

[2] J. Mitola, *Cognitive Radio: An Integrated Agent Architecture for Soft-ware Defined Radio*, Ph. D. Thesis, KTH, Nov. 2000.

[3] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Selected Areas in Communications*, vol. 23, no. 2, pp. 201-220, Feb. 2005.

[4] S. Fischer, M. Petrova, P. Mähönen, and B. Vöcking, "Distributed load balancing algorithm for adaptive channel allocation for cognitive radios," *Proc. IEEE Int. Conf. Cognitive Radio Oriented Wireless Networks and Communications*, pp. 508-513, Aug. 2007.

[5] L. Berlemann and B. Walke, "Spectrum load smoothing for optimized spectrum utilization-rationale and algorithm," *Proc. IEEE Wireless Communication and Networking Conf.*, vol. 2, pp.

735-740, Mar. 2005.

[6] A. T. Chronopoulos, M. R. Musku, S. Penmatsa, and D. C. Popescu, "Spectrum load balancing for medium access in cognitive radio systems," *IEEE Commun. Lett.*, vol. 12, no. 5, pp. 353-355, May 2008.

[7] S. Penmatsa and A. T. Chronopoulos, "Dynamic multi-user load balancing in distributed systems," *Proc. IEEE Int. Parallel and Distributed Processing Symp.*, pp. 1-10, Mar. 2007.

[8] S. Penmatsa, A. T. Chronopoulost, "Price-based user-optimal job allocation scheme for grid system," *Proc. IEEE Int. Parallel and Distributed Processing Symp.*, pp. 396-403, Apr. 2006.

[9] S. Mangold, *Analysis of IEEE 802.11e and Application of Game Models for Support of Quality-of-Service in Coexisting Wireless Networks*, Ph. D. Thesis, ComNets, RWTH Aachen University, Nov. 2003.

[10] 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.

Appendix I

Notations referring to [7] and additional parameter identifications:

For DSLB algorithm:

α_i^j : Mean interarrival time of a device j to channel i.

ϕ_i^j : Mean event access rate of device j to channel i.

ϕ^j: Total event access rate of the device j. $\phi^j = \sum_{i=1}^{n} \phi_i^j$.

Φ : Total event access rate of the system. $\Phi = \sum_{j=1}^{m} \phi^j$.

β_i : Mean occupying time of an event at channel i.

μ_i : Capacity bound of channel i. $\mu_i = \frac{1}{\beta_i}$

λ_i^j : Event processing rate (load) of device j at channel i.

$\lambda_i = [\lambda_i^1, \lambda_i^2, ..., \lambda_i^m]^T$, is the vector of loads at channel i from devices $1,...,m$.

$\lambda = [\lambda_1, \lambda_2, ..., \lambda_n]^T$, is the load vector of all channels from $1,...,n$.

$\lambda^k = [\lambda_1^k, \lambda_2^k, ..., \lambda_n^k]^T$, is the vector of loads of device k at channels $1,...,n$.

t_{rs}^j : Rate of device j from channel r to s.

T^j : Traffic through the system of device j.

$T^j = \sum_{r=1}^{n} \sum_{s=1}^{n} t_{rs}^j$.

N_i^j : Mean number of events of device j at channel i.

n_i^j : Number of events of device j at channel at a given instant.

t : Mean hand off time from one channel to another for any device.

For SLBDH algorithm:

w^j : Each demanded time length of each device j.

μ'_s : The available length of each slot.

l : Intersected granularity of short time cells of demanded time length.

ξ^j : The number of time cells of device j.

K_p^j : Hash key number of time cell p for device j. (p=1,...,ξ^j)

S_p^j : Mapped slot number of K_p^j.

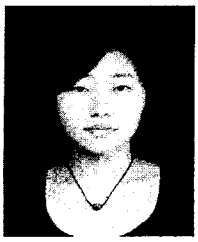
W_s : Weight of each slot. ($s = 1,...,n'$)

L_s : Aggregate load of each slot.

L_{ds}: Dynamic load of slot.

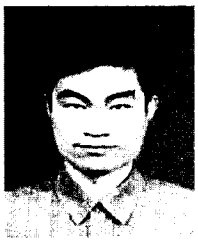
L_{fs}: Fixed load of PU and other occupied load by SU.

저자소개



Chen, Juan
2006년 7월 : 중국 남경체신대학교 전자정보공정 공학사
2007년 9월 : 중국 중경체신대학교 통신정보공정학원 공학석사
2008년 9월 ~ 현재 : 인하대학교 정보통신대학원 공학석사

손 성 환 (Sohn, Sung-Hwan)
2004년 2월 : 인하대학교 전자공학부 공학사
2006년 2월 : 인하대학교 정보통신대학원 공학석사
2006년 3월~현재 : 인하대학교 정보통신대학원 공학박사



Gu, Junrong
2005년 7월 : 중국 동북대학교 통신공정 공학사
2008년 1월 : 중국 대련이공대학교 통신정보계통 공학석사
2008년 3월 ~ 현재 : 인하대학교 정보통신대학원 공학박사

김 재 명 (Kim, Jae-Moung)
1974년 2월 : 한양대학교 전자공학과 공학사
1981년 8월 : 미국 남가주대학교(USC) 전기공학과 석사
1987년 8월 : 연세대한교 전자공학과박사
1974년 3월 ~ 1979년 6월 : 한국과학기술연구소, 한국통신기술연구소 근무
1982년 9월~ 2003년 3월 : 한국전자통신연구원 위성통신연구단장/무선방송연구소 소장역임
2003년 4월 ~ 현재 : 인하대학교 정보통신대학원 원장/교수, 현재 통신위성 우주산업연구회 회장 외 기술자문으로 다수 활동 중