Utilizing the Inherent Properties of Preamble Sequences for Load Balancing in Cellular Networks

Ankit Chopra #1, Peter Sam Raj *2, Winston K.G. Seah #3

School of Engineering and Computer Science, Victoria University of Wellington
Wellington, New Zealand
1 ankit.chopra@ecs.vuw.ac.nz
3 winston.seah@ecs.vuw.ac.nz

* *Indian Institute of Technology*
Guwahati, India
2 peter@iitg.ernet.in

***Abstract*—Achieving maximum network throughput while providing fairness is one of the key challenges in wireless cellular networks. This goal is typically met when the load of base stations (BSs) is balanced. In a recent study, it was confirmed that the preamble sequence, which is mainly used for cell identification and synchronization, can be used as an implicit load indicator for long term load balancing in cellular networks. In this paper, we further explore its viability as an implicit information indicator, and present distributed load balancing schemes based on the properties of the preamble sequence. These schemes empower the mobile stations (MSs) to implicitly obtain information about the load status of BSs for intelligent cell-site selection. By utilizing the two characteristic and important properties of preamble sequences, which are high auto-correlation and low cross-correlation, we propose both network and user side algorithms that work together to enable the MSs to associate with BSs with relatively low load in the long term. We evaluate our algorithms for fairness to the MSs in terms of resources and throughput, and confirm through simulations that the preamble sequence can indeed be used as a leverage for better cell-site selection leading to a high degree of fairness in the system.**

***Index Terms*—Preamble, load balancing, fairness.**

I. Introduction

In a cellular network, a mobile station (MS) generally scans and associates itself with the base station (BS), which yields the maximal received signal strength indicator (RSSI) obtained by reading the power of the preamble signal, while being oblivious to the load of the BS. As MSs are, typically, not evenly distributed, some BSs tend to suffer from heavy load while their adjacent BSs may carry only light load. Such load imbalance severely hampers the network from fully utilizing the network capacity and providing fair services to users. To address this, various load balancing schemes have been proposed for efficient utilization of resources that enable the users to enjoy better throughput and operators to support more subscribers. Most of the load balancing schemes that have been proposed are based on centralized control by the network, and often suffer from the problem of scalability due to signalling overhead and their focus on balancing short-term load variation [1]-[4]. Therefore, in this paper we emphasize mainly on the distributed mechanism wherein the MS controls the association based on our implicit load indication schemes using the preamble sequence. It is also worth mentioning that, since we want to balance long-term load variation, our schemes can be employed along with any other short-term load balancing schemes, and do not suffer from the capacity reduction problem that is seen in schemes relying on power control [1]-[2].

II. Related work

In a recent study by Na, *et al.* [5], it has been shown that preamble sequence can be used as an implicit load indication tool. The work exploited comparative high and low cross correlation among certain sets of preamble sequences, to indicate similarity in terms of lightly loaded and heavily loaded state of the BSs for cell selection. However, the proposed scheme fails to address all the permutations of BS loads, and in situations where certain PN sequences outside the set would yield high correlation with sequences within it. We, therefore, propose an alternative approach to using preamble sequences for load indication by making the implicit indication absolute, and thereby, removing the inherent difficulties in a relative scheme.

The rest of the paper is organized as follows. In section II, we describe how preamble sequences can be configured to implicitly obtain the load status of BSs using our proposed auto-correlation property based, and absolute cross-correlation based schemes. In section III, we define our system model, and use the IEEE 802.16e [6] for illustration purposes. After that, in section IV, we present our proposed algorithms with their pseudo-codes. In section V, simulation results are presented to validate our schemes and finally in section VI, we conclude the paper with a discussion of future work possible in this direction.

III. PREAMBLE SEQUENCE FOR LOAD BALANCING

A preamble sequence is essentially a pseudo-noise sequence (PN sequence) that is used for time and frequency synchronization, channel measurement, and cell identification in wireless networks. They are called PN sequences because of their low autocorrelation for positive delays. This makes them nearly random, even though they are deterministic in nature. They are useful in cell identification because of their uniqueness, i.e. different PN sequences are highly uncorrelated with each other, but the cross correlation of a PN sequence with an identical copy of itself produces a peak at the origin. In this paper, we configure them to indicate the load experienced by their BSs, and thus allowing the MSs to make decisions which are more efficient in the fairness sense, depending on the QoS measure we take as a criteria. We use the properties of high autocorrelation and low cross-correlation to define certain metrics (different in the two schemes proposed), which can be used to sort the preamble sequences. Then, we implicitly link the load information to the value of the metric associated with the preamble sequence by mapping the preamble sequences to the BSs according to the value of the metric and the load experienced by the BS. For example, for a highly loaded BS b, the corresponding metric defined on b's preamble sequence $\overrightarrow{p}_b$, would have a higher value and vice-versa. Let us consider the following preamble configurations:

- A sequence $\overrightarrow{p}$ formed by repetition of a part of a preamble sequence on autocorrelation forms a number of peaks in the autocorrelation plot which will be proportional to the load experienced by the BS corresponding to this sequence $\overrightarrow{p}$.
- A preamble sequence of a lightly loaded BS will be highly correlated with a particular code sequence $\overrightarrow{p}_0$ (known to both the MS and the controller) and vice-versa.

In each of the above mentioned configurations, we propose two algorithms which work in parallel at the MS and at the central network controller. For the algorithm to implement the load balancing feature, we propose a design parameter r which is basically a measure of how much greater the strongest RSSI is over the next strongest RSSI in the system, when our algorithm stops coming into effect. This physically implies the measure of area near the border between two cells where load balancing comes into effect. As r increases, more area near the boundary is considered and vice-versa. Here, $r = 1$ will be the limiting case, where the ratio

$$\alpha = \frac{RSSI_{strongest}}{RSSI_{nextstrongest}} = 1 \quad (1)$$

occurs at the cell boundary. From this limiting case onwards, when $r > 1$, larger areas near the boundary are considered for load balancing. Additionally, the conventional RSSI based cell selection can be invoked by setting $r = 0$, when the algorithm skips the load balancing steps and adopts the conventional method using RSSI since α is always positive. Thus, when $r = 0$, the condition for $\alpha < 0$ will never be satisfied as can be seen later in the MSCD_AutoCorr and MSCD_CrossCorr algorithms, and can be better understood from Fig. 1. Using the above described metrics, the sequences corresponding to the BSs are sorted and assigned in a predefined order as shown in Fig. 2 for the auto-correlation based load indication scheme and Fig. 3 for absolute cross-correlation scheme.

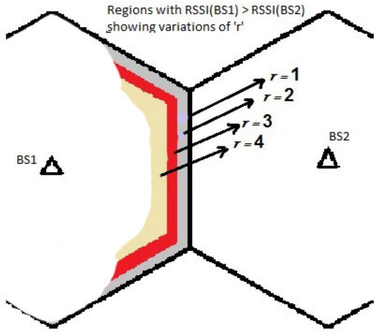

Fig. 1. A two cell model showing the concept of "r".

IV. SYSTEM MODEL

For illustration purpose, we will consider the IEEE 802.16e specification [6], which defines 114 pseudo noise (PN) codes to be used as the preamble sequences in the 1k Fast Fourier Transform mode. These Preamble sequences are 284 bits long, and have the characteristic properties that make them appealing for applications like cell identification and synchronization. In this paper, we work with decisions involving cross-correlation between the PN code sequences, as the average cross-correlation between the different PN code sequences have been found to be ranging from a minimum of 0.04 to a maximum of 0.16 [5].Also, a recent study [7] shows, it is possible to use just a part of the whole PN code sequence to practically achieve the goals of synchronization and cell identification, mainly because of this cross-correlation property among different sequences.

In this paper, we adopt a typical cellular network model (cf: Fig. 5) where the MS generally decides on choosing BS's based on the Received Signal Strength Indicator (RSSI). In a model where $\mathcal{B}$ is the set of all the base stations, g_{bs} is the channel gain from BS b to an MS s, P_b is the transmission power of BS b, $\overrightarrow{p}_b$ is the preamble code sequence corresponding to base station b, and $\overrightarrow{x}_s(t)$ is the signal received by an MS s at time t from all BSs; the decision

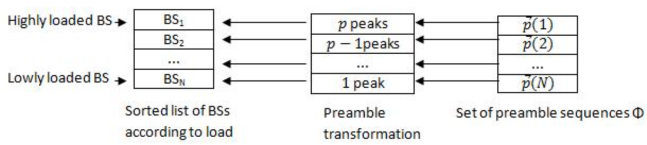

Fig. 2. At central controller the metric is number of peaks in the auto-correlation plot.

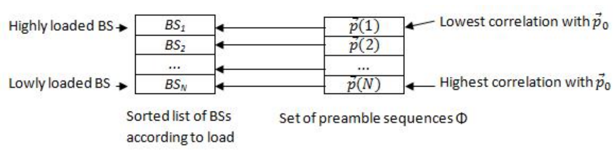

Fig. 3. At central controller the metric is value of cross-correlation with particular sequence.

made by the MS s is given in Eq. 2, where $\hat{b}$ is the chosen BS.

$$\hat{b} = arg \max_{b \in \mathcal{B}} RSSI_s(t,b) \quad (2)$$

$$RSSI_s(t,b) = \overrightarrow{p}_b . \overrightarrow{x}_s(t) \quad (3)$$

$$\overrightarrow{x}_s(t) = \sum_{b \in \mathcal{B}} g_{bs} P_b(t) \overrightarrow{p}_b \quad (4)$$

where RSSI refers to the signal component of base station b, and $\overrightarrow{p}_b \in \mathcal{P}$ refers to the preamble sequences, such that $\mathcal{P}$ is the set of all BS preamble sequences. In addition, the k^{th} sequence in the set can be referred to as $\overrightarrow{p}(k)$ and will be used in the later sections.

V. PROPOSED ALGORITHMS

This section describes, with the help of pseudo codes, the two algorithm pairs proposed for BS load balancing.

A. Use of the Auto-correlation property

We propose two algorithms working side by side to provide BS load indication using the preamble sequence. In this method, we modify the preamble code sequence and propose the autocorrelation operation as a way to decipher the load in a BS. This approach works at two levels: at the MS and at the central network controller.

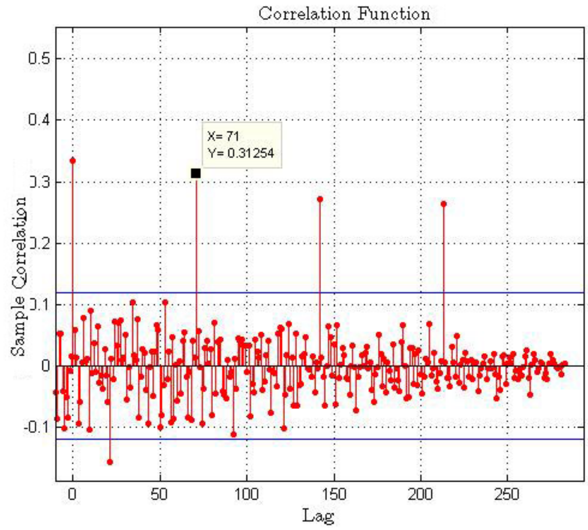

Fig. 4. Auto-correlation plot of a preamble of length 284 bits made with 4 repeated sequences of length 71.

Here, $\mathcal{B}$ is the set of all BSs and a particular BS is b with load l_b. First, the central controller creates a list of BSs sorted in order of their loads. Then, we define p intervals of the load value which can be denoted by $L_1, L_2, ...L_p$. Thus, the load values l_b pertaining to the base stations b fall in one of these ranges and are grouped into one of L_j. In a general case where the preamble sequence length is M bits, the autocorrelation plot of the sequence will give a peak at the origin and negligible values for all other delays. This is a characteristic feature of the PN sequences due to their pseudo random nature. In this paper, we modify the PN sequence by selecting the first $\frac{M}{j}$ bits of the PN sequence and then repeating it j times where $j = arg\{L_j\}$. This will give us a sequence of M bits which would be periodic in the time domain and would have j periods in the M bit length.

After this operation, the autocorrelation plot for the modified preamble code shows j peaks corresponding to the number of repetitions in the M bit sequence. We propose that this value j be the metric which we use for load indication of a particular BS. Hence, more the repetitions in the preamble sequence, more will be the number of peaks in the autocorrelation plot and that would indicate a more loaded BS. This modification to the preamble code sequence by introducing periodicity in the time domain does cause concerns over its effect on cell identification and synchronization which the preamble sequence actually has to fulfil. These are valid concerns as due to the periodic nature of the modified preamble, it exhibits lower separation of magnitude between the autocorrelation peak and other values on the autocorrelation plot. These concerns have been addressed in a recent study [7] which shows that a considerably small part of the preamble sequence is practically sufficient in terms of achieving the required goals of identifiability and also synchronization. Larger number of repetitions would make the errors higher, but below a maximum number of allowed repetitions p, it would be practically sufficient.

Preamble Modifying Algorithm: The modification of the preamble code would happen at the central network controller where it will assign the j value (number of repetitions) to the BSs according to their loads. This will be done by first identifying the maximally loaded BS in the set of BSs $\mathcal{B}$ (line 3). Then, the group number of the load value is extracted (line 4, 5) and accordingly, the preamble sequence is modified (line 7 - 9) by taking the first $\frac{M}{j}$ bits of $\overrightarrow{p}$ and repeating it j times (line 8). Thus, the BSs will then transmit using preamble sequences that have j repetitions of the first $\frac{M}{j}$ bits of the original sequence. This we call as the PMA or Preamble Modifying Algorithm.

Pseudo-code for PMA
1 **procedure** PMA($p, \mathcal{B}$)
2 **while** $\mathcal{B} \neq \emptyset$ **do**
3 $\acute{b} \longleftarrow arg \max_{b \in \mathcal{B}} l_b$
4 **if** $l_{\acute{b}} \in L_j$ **then**
5 $j \longleftarrow arg\{L_j\}$
6 **endif**
7 $\overrightarrow{p} \longleftarrow \overrightarrow{p}_{\acute{b}}$ where $\overrightarrow{p} \in \mathbb{R}^M$
8 $\overrightarrow{p} \longleftarrow [\overrightarrow{p}(\frac{M}{j})\overrightarrow{p}(\frac{M}{j})...\overrightarrow{p}(\frac{M}{j})]$ (repeated j times)
9 $\overrightarrow{p}_{\acute{b}} \longleftarrow \overrightarrow{p}$
10 $\mathcal{B} \longleftarrow \mathcal{B}/\{\acute{b}\}$
11 **end while**
12 **end procedure**

MS Cell Decision using AutoCorrelation: Then, at the MS, the MS Cell Decision (MSCD) algorithm will be used to detect load values of BSs which are of interest to the MS and consequently make cell choice decision. We denote this MSCD algorithm using auto-correlation as MSCD_AutoCorr. The MS first correlates the preamble code with the copy of the original preamble sequence code list it possesses to identify the BSs (this step is involved in detecting the three strongest BS signals in lines (2 - 6) of MSCD_AutoCorr) and then uses the autocorrelation property (line 8 - 10) to find multiple peaks (if applicable) instead of the single peak which would have been the normal case. Detecting the number of peaks (line 11-13), it arrives at the value j which conveys information about the load, which is then used to decide on the choice between the three BSs of interest. Prior to the Preamble Modifying Algorithm, the grouping intervals of the load values are assumed to have been decided by the central network controller using uniform grouping.

The $peakfind()$ function finds the number of peaks in the argument plot for different values of τ and returns a value which is equal to the number of peaks in it. Here we define two design parameters r and δ, which specifies the area to be influenced by load balancing algorithm and the minimum RSSI requirement, respectively.

Pseudo code for the MSCD_AutoCorr Algorithm
1 **procedure** MSCD_AutoCorr
2 $\overrightarrow{p}(k_1) \longleftarrow arg \max_{\overrightarrow{p}(k)\in\mathcal{P}} RSSI_s(\overrightarrow{p}(k))$
3 $\mathcal{P} \longleftarrow \mathcal{P}/\{\overrightarrow{p}(k_1\}$
4 $\overrightarrow{p}(k_2) \longleftarrow arg \max_{\overrightarrow{p}(k)\in\mathcal{P}} RSSI_s(\overrightarrow{p}(k))$
5 $\mathcal{P} \longleftarrow \mathcal{P}/\{\overrightarrow{p}(k_2\}$
6 $\overrightarrow{p}(k_3) \longleftarrow arg \max_{\overrightarrow{p}(k)\in\mathcal{P}} RSSI_s(\overrightarrow{p}(k))$
7 **if** $\frac{RSSI_s(\overrightarrow{p}(k_1))}{RSSI_s(\overrightarrow{p}(k_2))} < r$ && $RSSI_s(\overrightarrow{p}(k_2)) > \delta$, **then**
8 $a(k_1,\tau) \longleftarrow autocorr(\overrightarrow{p}(k_1)$ where τ is delay
9 $a(k_2,\tau) \longleftarrow autocorr(\overrightarrow{p}(k_2)$
10 $a(k_3,\tau) \longleftarrow autocorr(\overrightarrow{p}(k_3)$
11 $j(k_1) \longleftarrow peakfind(a(k_1,\tau))$
12 $j(k_2) \longleftarrow peakfind(a(k_2,\tau))$
13 $j(k_3) \longleftarrow peakfind(a(k_3,\tau))$
14 $\acute{k} \longleftarrow arg \min_{k\in\{k_1,k_2,k_3\}} \{j(k)\}$
15 $\{b_i^*\} \longleftarrow \{b \mid \overrightarrow{p}_b = \overrightarrow{p}(\acute{k})\}$
16 **else**
17 $\{b_i^*\} \longleftarrow \{b \mid \overrightarrow{p}_b = \overrightarrow{p}(k_1)\}$
18 **end if**
19 **end procedure**

B. Use of the absolute cross-correlation property

In this approach, what we call the *Preamble Allocation using Cross-correlation (PAC)* algorithm is performed at the centralized controller. Here we use the maximum correlation value of the preamble with a particular predefined sequence $\overrightarrow{p}_0$. This sequence can be an arbitrary preamble sequence taken from one of the 114 PN codes which are used in the 1k FFT mode in IEEE 802.16e. When we correlate the BS's preamble code with this particular code $\overrightarrow{p}_0$, the result can then be used as a metric to which we can implicitly link the load of the BS. This is largely dependent on getting different such cross-correlation values for different preamble sequences. In this method, the set Φ_s contains the sorted list of various preamble sequences sorted with respect to their cross-correlation value with $\overrightarrow{p}_0$. In Φ_s, the first preamble sequence has the lowest cross-correlation with $\overrightarrow{p}_0$ and the last one has the highest cross-correlation with $\overrightarrow{p}_0$.

Before the commencement of this algorithm, the central network controller will obtain the load values of all the base stations. This load could be the number of users connected or the number of calls connected. After this, the controller would create a list Φ_s which would contain the preamble sequences sorted according to the value of their maximum cross-correlation with the reference preamble sequence $\overrightarrow{p}_0$. The algorithm then searches the most loaded BS in the set $\mathcal{B}$ (line 4). It then assigns the first sequence in Φ_s, which is the one with the least correlation with $\overrightarrow{p}_0$ to it (line 5). Then, the pointer to the top of the list Φ_s is incremented to the sequence with the next lowest correlation with $\overrightarrow{p}_0$ and the whole process is repeated till all the BSs are exhausted.

Pseudo-code for the PAC Algorithm
1 **procedure** PAC($\mathcal{B}$)
2 $i = 1$
3 **while** $\mathcal{B} \neq \emptyset$ **do**
4 $\acute{b} \longleftarrow arg \max_{b\in\mathcal{B}} l_b$
5 $\overrightarrow{p_{\acute{b}}} \longleftarrow \overrightarrow{p}(i)$ s.t. $\overrightarrow{p}(i) \in \Phi_s$
6 $i \longleftarrow i+1$
7 $\mathcal{B} \longleftarrow \mathcal{B}/\{\acute{b}\}$
8 **end while**
9 **end procedure**

MSCD using Cross-Correlation algorithm: Together with the PAC algorithm, the MSCD using Cross-Correlation (denoted as MSCD_CrossCorr) algorithm is employed by the MS if it is near the boundary between three BSs. Firstly the algorithm identifies the three nearest BSs by using their RSSI (line 2 - 6). Then, a condition is tested using a design parameter r which determines whether the MS is at the boundary area between the two identified BS's cells by calculating the ratio of their RSSI.

Pseudo code for the MSCD_CrossCorr Algorithm
1 **procedure** MSCD_CrossCorr
2 $\overrightarrow{p}(k_1) \longleftarrow arg \max_{\overrightarrow{p}(k)\in\mathcal{P}} RSSI_s(\overrightarrow{p}(k))$
3 $\mathcal{P} \longleftarrow \mathcal{P}/\{\overrightarrow{p}(k_1\}$
4 $\overrightarrow{p}(k_2) \longleftarrow arg \max_{\overrightarrow{p}(k)\in\mathcal{P}} RSSI_s(\overrightarrow{p}(k))$
5 $\mathcal{P} \longleftarrow \mathcal{P}/\{\overrightarrow{p}(k_2\}$
6 $\overrightarrow{p}(k_3) \longleftarrow arg \max_{\overrightarrow{p}(k)\in\mathcal{P}} RSSI_s(\overrightarrow{p}(k))$
7 **if** $\frac{RSSI_s(\overrightarrow{p}(k_1))}{RSSI_s(\overrightarrow{p}(k_2))} < r$ && $RSSI_s(\overrightarrow{p}(k_2)) > \delta$, **then**
8 $cc(k_1) \longleftarrow crosscorr(\overrightarrow{p}(k_1), \overrightarrow{p}_0)$
9 $cc(k_2) \longleftarrow crosscorr(\overrightarrow{p}(k_2), \overrightarrow{p}_0)$
10 $cc(k_3) \longleftarrow crosscorr(\overrightarrow{p}(k_3), \overrightarrow{p}_0)$
11 $\mathcal{Q} \longleftarrow \{\overrightarrow{p}(k_1), \overrightarrow{p}(k_2), \overrightarrow{p}(k_3)\}$
12 $\acute{k} \longleftarrow arg \max_{k=arg \overrightarrow{p}(k) s.t. \overrightarrow{p}(k)\in\mathcal{Q}} cc(k)$
13 $\{b_i^*\} \longleftarrow \{b \mid \overrightarrow{p}_b = \overrightarrow{p}(\acute{k})\}$
14 **else**
15 $\{b_i^*\} \longleftarrow \{b \mid \overrightarrow{p}_b = \overrightarrow{p}(k_1)\}$
16 **end if**
17 **end procedure**

Also, a minimum RSSI criteria is tested using a parameter δ (line 7). When either of these two conditions fail, the usual

RSSI based selection is followed (line 15). Otherwise, the load balancing algorithm comes into effect. Cross correlations of the preamble sequences of the three BSs are calculated (line 8 - 10) and the BS having highest maximum cross-correlation is chosen.

VI. Simulation results

In this section, we verify the applicability of using the preamble sequences to indicate load information to the MS through simulations based on MATLAB. In this regard, we consider a 12 cell model with a frequency reuse factor of 1, as illustrated in Fig. 5, and focus on the three cells in the center with red, blue and green MSs. For the initial distribution of users, we assume that 100 stations are located in cell 0, 80 stations are located in cells 1, 20 stations are located in cell 2 and all the other cells have 45 stations randomly distributed over their cell region. Our algorithm is then applied to the boundary stations. Also, for the sake of simplicity, we only consider the channel gain due to path loss effect with a path loss exponent of 5.

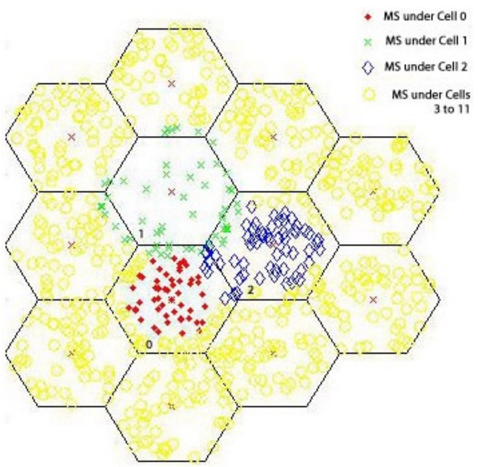

Fig. 5. Distribution of users in cells at r = 10. Note that the initial high load of Cell 0 with 100 users has now changed and it has handed over many of its MSs to the adjacent BSs after the algorithm runs for r = 10, and they show now change of BS by changing colour. Hence, the BS becomes less heavily loaded. Similarly observe Cell 2 with initially 80 users and finally much lesser, on account of handing over of MSs to lightly loaded adjoining cells.

In distributed systems, fairness is an important performance criterion for allocation of resources or services. It is essentially equal allocation of resources or features. In this paper, we take those features to be bandwidth, which is a limited resource and throughput, in terms of minimum data rate available equally to all users. As the major target of every load balancing scheme are cell boundary users, more users participate in load balancing as r increases.

Simulations in this section implement the proposed algorithms at the decision making level where depending on the load of BSs, the stations in a certain area (defined by r) near the boundary choose the BS to be associated with. Hence, as both algorithms look at the same quantity i.e. load to help MS's make the cell choice, the simulation results for them are identical. The differences in the two algorithms lie in the method of detection of load information in the preamble, and thus when this is accurately done as assumed in the simulations, both algorithms give identical results. Fig. 6 shows that the number of users becomes more evenly distributed as r increases. This can be visually seen as in Fig. 5 where the red MSs are associated to heavily loaded cell 0, blue MSs are associated to cell 2 with heavy load again and the green MSs are associated with the lightly loaded cell 1. At $r = 10$ case, one can see how our algorithm leads many of the MSs near the boundary of cell 0 to be handed over to neighbouring BSs and thus becoming green/blue/yellow, as cell 0 is heavily loaded. Similarly, blue MSs are also handed over to cells depending on their loads. Green MSs however are mostly those who have been handed over to BS of cell 1 by other heavily loaded BSs. Thus, as we see in Fig. 6, at $r = 10$, all three BSs of cells 0 to 2 have number of associated MSs between 50 to 60. This can be interpreted that our load balancing scheme leads to an improved consequence in terms of fair resource sharing.

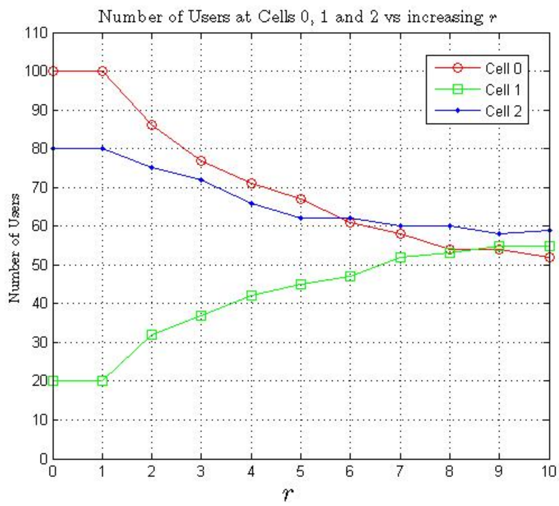

Fig. 6. Number of Users in Cells 0 to 2 vs r

For minimum throughput simulations, we consider the lower bound case for throughput in a cell that occurs when all the users are at the maximum distance from the BS, and hence, have the lowest possible SNR at the MS. In such cases, the bandwidth is equally distributed among the MSs, but the capacity reaches its minima because of lowest SNR. This is clear from the expression for channel capacity [8], where R is the channel capacity, W is the bandwidth assigned to each MS, P is the power received by the MS and $N_0/2$ is the noise.

$$\frac{R}{W} = \log(1 + \frac{P}{N_0 W})$$

Thus, this lower bound of throughput in a cell is calculated for different variations of r and it is seen that the minimum throughput improves considerably for the heavily loaded BS

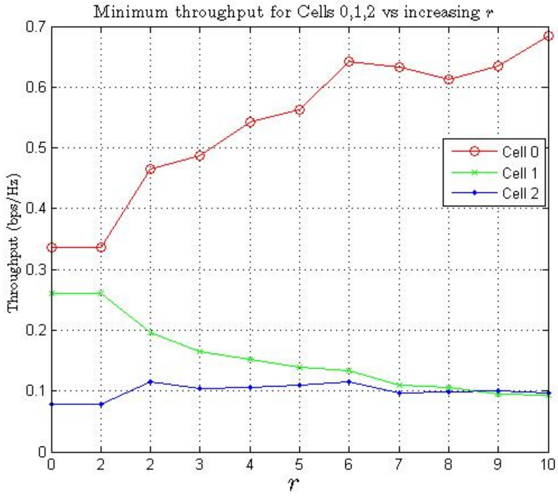

Fig. 7. Throughput in Cells 0 to 2 vs r

on application of the proposed algorithms for load balancing. This is because the initially heavily-loaded BS hands over some of its boundary users to the adjoining cells and hence, achieves better throughput. On the other hand, the other two lightly loaded BS's throughput deteriorates slightly because of the increase in number of users managed by the BS. This can be interpreted that our load balancing scheme leads to an improved consequence in terms of max-min throughput fairness.

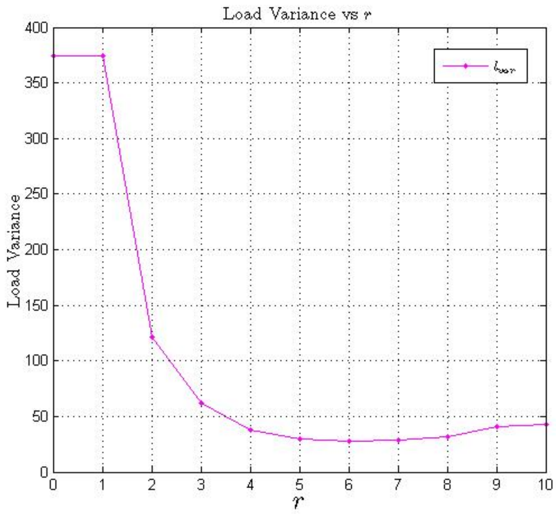

Fig. 8. Load Variance per BS vs r

As a quantitative measure of how the load is distributed among the various BS's cells, we propose a metric called the Load Variance l_{var} which is given by:

$$l_{var} = \frac{\sum_{b \in \mathcal{B}} (l_b - \bar{l}_b)^2}{n(\mathcal{B})}$$

where $n(\mathcal{B})$ is the total number of BSs, l_b is the load experienced by each BS and $\bar{l}_b$ is the mean of all the load values. Therefore, as the load gets distributed more evenly among the different BSs, we see a gradual decrease in l_{var}, as seen in Fig. 8, which shows its variation with different values of our design parameter r. Thus, this metric is useful in measuring the even-ness of distribution of MSs in a collection of cells.

VII. Conclusion and Future Work

In this paper, we have confirmed the applicability of the preamble sequence as an implicit information indicating tool and stations can take advantage of this in selecting a more appropriate BS, thereby maintaining resource and throughput fairness across the network. In the future, we shall work and try to concatenate repeated PN sequences to form preambles in such a way that would effectively minimize the degradation of the preamble sequence in terms of detection. We shall also work on developing methods that are computationally less complex involved in cross-correlation of preamble sequences.

References

[1] R. Yates and C.-Y. Huang, Integrated power control and base station assignment, IEEE Trans. on Veh. Technol., vol. 44, no. 3, pp. 638-644, Aug. 1995.

[2] S. V. Hanly, "An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity," *IEEE J. Select. Areas in Comms.*, vol. 13, no. 7, pp.1332-1340, Sep 1995.

[3] Aimin Sang, *et al.*, "Coordinated load balancing, handoff/cell-site selection and scheduling in multi-cell packet data systems," in *Proc. ACM MobiCom*, Philadelphia, PA, USA, Sep 26-Oct 1, 2004.

[4] S. Das, *et al.* "Dynamic load balancing through coordinated scheduling in packet data systems," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr 1-3, 2003.

[5] Minsoo Na, *et al.*, "On Utilizing Preamble Code Sequence as Implicit Load Indicator," in *Proc. IEEE APWCS*, Sendai, Japan, Aug 21-22, 2008.

[6] IEEE 802.16e-2006, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems-Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, IEEE std., Feb. 2006.

[7] Peng Cheng, *et al.*, "A study on cell search algorithms for IEEE 802.16e OFDMA systems," in *Proc. IEEE WCNC*, Hong Kong, Mar 11-15, 2007.

[8] Shannon, *et al.*,"Communication in the presence of noise," in *Proc. IEEE*, Sept ,1984.