Gap-free Load Balancing Algorithms in Wireless LANs using Cell Breathing Technique

İlhan DEMİRCİ [#1] , Ömer KORÇAK [#2]
[#] *Department of Computer Engineering, Marmara University, İstanbul, Turkey,*
[1] `ilhandemirci@marun.edu.tr`, [2] `omer.korcak@marmara.edu.tr`

***Abstract*—A balanced network generally has a fair user distribution among the access points (APs) and enables a maximized overall network throughput. However, there are some hot-spot areas resulting in uneven load distribution among APs. Existing load balancing techniques mostly require hardware or software contribution at user side to control the association of users with APs. Power adjustment techniques at AP side require no modification at user devices, but, coverage reduction can create unserviced areas. We provide a dynamic min-max load balancing algorithm based on power adjustment technique which provides service availability guarantee by dynamically checking the coverage holes. We enhanced the algorithm by increasing awareness to heterogeneous nature of user distribution. We show the effectiveness of the provided algorithms via extensive set of simulations. Simulation results show that the algorithms improved the load balancing among the hot-spot areas up to 10%.**

I. Introduction

Recently, Wireless Local Area Networks (WLANs) deployed enormously in many public places such as university campuses. Users having mobile devices can connect to the Internet using IEEE 802.11 protocol. Uneven user distribution is the common characteristics of these types of networks. This results in overloading problem in some of the access points (APs), although their adjacent APs may carry only light load. Clearly such a load imbalance is undesirable since it reduces utilization of network capacity and prevents from providing fair services to users. Providing a load balancing solution to the WLAN protocol is of paramount importance.

There are several approaches proposed for load balancing in WLAN networks. Most of them rely on software/hardware modifications in the client side. One promising approach is the cell breathing technique which doesn't necessitate any change in users' equipment. In the cell breathing technique, access points change their coverage areas by modifying their beacon signal strengths. When an AP decreases its beacon signal strength, its coverage area shrinks which looks like exhaling in the human breathing. Inversely, by increasing the beacon signal strength, an AP can enlarge its coverage area which resembles inhaling. The mechanism behind the cell breathing is to drop some users from connecting to the AP by decreasing beacon signal strength when the cell becomes heavily loaded. Dropped users will then connect to the neighboring cells that are more lightly loaded.

Our work is based on cell breathing technique. Our first concern is to provide load balancing and system utilization with sparse wireless AP deployment where coverage area shrinking can result in uncovered areas among them. For this purpose, we extend the min-max load balancing algorithm stated in [3]. Secondly, we propose a method to define minimum permissible beacon power levels by using statistical network usage information. Lastly, we present an online min-max load balancing solution that determines the minimum permissible power levels by considering the current load values.

A. Related Work

Recent studies show that, in most of the WLANs, majority of connections are generated through small number of APs [9]. 802.11 protocol does not provide any mechanism to balance the load in a congested network. Hence, researchers proposed techniques to manage the network load. These techniques can be classified as *client-supported* and *network-controlled* load balancing mechanisms.

In client-supported mechanisms, clients actively measure [8], [6] or passively learn [5], [13], [12] the loads of APs and use these load information together with the received signal strength indicator (RSSI) in order to decide on which AP to use. In [8], Vasudevan et al. propose an AP selection method based on potential bandwidth estimation that relies on passive measurements of the timings of AP beacon frames. Nicholson et al. [6] propose an AP association method, where mobile devices quickly associate to each available AP and test their connection quality before selecting the most suitable one. Yukuda et al [13] and Gong et al. [5] present distributed decentralized algorithms for AP selection, where mobile devices use AP load information which they receive from AP beacons and probing packets. Yen et al. [12] propose deployment of a dedicated server which collects load-based information from APs utilizing Simple Network Management Protocol (SNMP) and clients contact this server to learn the most appropriate AP that maximizes overall system throughput.

One significant disadvantage of mentioned client supported schemes is that they require software or hardware modifications at the client side. On the other hand, network controlled schemes employ a network-side entity to control the load of APs, and can achieve load balancing without any change in user equipment. Coverage adjustment algorithms proposed in [1], [3] and [11] fall into this class. Bahl et al. [1] proposed a cell breathing algorithm to maximize the overall network throughput. They formulate cell breathing problem as mixed integer linear programming, assuming continuous beacon power levels. A later work by Bejerano et al. also considers cell breathing approach and aim to achieve min-max fair load balancing [3]. Different than Bahl et al., they assume that only discrete set of beacon power levels are feasible, which can be considered as more practical assumption. In [3], it is assumed that APs are located without causing gaps even if all the APs

TABLE I. ABBREVIATIONS STATED IN ALGORITHMS

Definition	Symbol	Definition	Symbol
Users	U	Fixed AP List	F
Access Points	AP	Access Point Locations	$APLoc$
Minimum Permissible Power Level of AP$_i$	μ_i	Number of Access Points	AP_{num}
Beacon Power Level of AP$_i$	p_i	Minimum Power Level	P_{min}
Rectangular region covered by APs	R	Maximum Power Level	P_{max}
Recorded Maximum Loaded AP Index	$\tilde{d}$	Load of AP$_i$	λ_i
Congestion Load of Recorded State	$\tilde{\Lambda}$	Maximum Loaded AP Index	d
List of Access Points	A	Congestion Load Value	Λ
Maximum Power Level Number	K		

transmit at minimum power level. By assigning the minimum permissible power levels using radio coverage survey method, they provided a static solution to coverage hole issue. Recently, Wang et al. [11] take the service availability problem into consideration and present an optimization problem for cell breathing technique that minimizes the variance of all AP's loads under full service coverage constraints. They propose a polyhedron genetic algorithm which provides a near optimal beacon range for each AP with a service availability guarantee. Their algorithm sets the beacon power levels in a continuous manner, as in the case of [1].

In this work, we consider min-max load balancing problem similar to the problem considered in [3], where only discrete set of power levels are available. However we also focus on full service availability problem, and provide gap detection and dynamic assignment of minimum permissible power levels considering the user distribution in the network region. The **main contributions** of this work are the following: *(i)* we extend min-max priority load balancing algorithm proposed by [3] such that minimum beacon power levels are assigned using the AP location information in order to satisfy full service coverage, *(ii)* we propose a novel algorithm that sets the minimum beacon power levels aware of the past load statistics, *(iii)* we define another algorithm that adjusts minimum beacon power levels dynamically according to the current load values, *(iv)* we show the effectiveness of proposed algorithms via extensive set of simulations.

The rest of the paper is organized as follows. In Section II we give the network model and define the problem and in Section III we describe the proposed algorithms. Next we include simulation results in Section IV and we conclude in Section V.

II. PROBLEM DEFINITION

A. Network Model

In this paper, we consider a WLAN network with multiple Access Points in a region R. AP regions are assumed to be circular. There are two signal levels for each AP; data transmission signal level and beacon signal level. Beacon signal level can be equal to or less then the data transmission signal level. We assume that with highest beacon signal level, the whole area R can be covered by APs. We provide a full data coverage area in region R where each point in the area is served by at least one AP. APs transmit the beacon signals in discrete manner. Each AP can use one of K+1 beacon power levels, denoted by $\{P_k \mid k \in [0..K]\}$, where the minimal and maximal levels are denoted by $P_{min} = P_0$ and $P_{max} = P_K$. We also define *minimum permissible power level*, $\mu_i \geq P_{min}$, which defines the minimum value of beacon power level that an AP can set without violating service availability. So, an AP i will be able to set its beacon power level, p_i, to the power levels that are higher than μ_i. On the other hand, APs transmit the data traffic with maximal power, i.e. data transmission signal level of an AP is always equal to P_{max}. In this environment, beacon signal adjustment of APs is handled by an Access Controller (AC), and communication between AP and AC can be based on a protocol like CAPWAP [4].

Figure 1 illustrates a small scenario with two APs serving to 20 users distributed over a rectangular area R. μ_1 and μ_2 values are set to 13 dBm and 16 dBm respectively. Any further decrease will violate full coverage of R, so APs are not permitted to set their beacon powers to less than these values. Beacon powers also cannot exceed P_{max}. Our concern in this work is to find best values of μ_i and p_i so as to satisfy load balancing as well as full service coverage. In Figure 1, AP 2 reduces its beacon power level more than AP 2, since there are more users close to it.

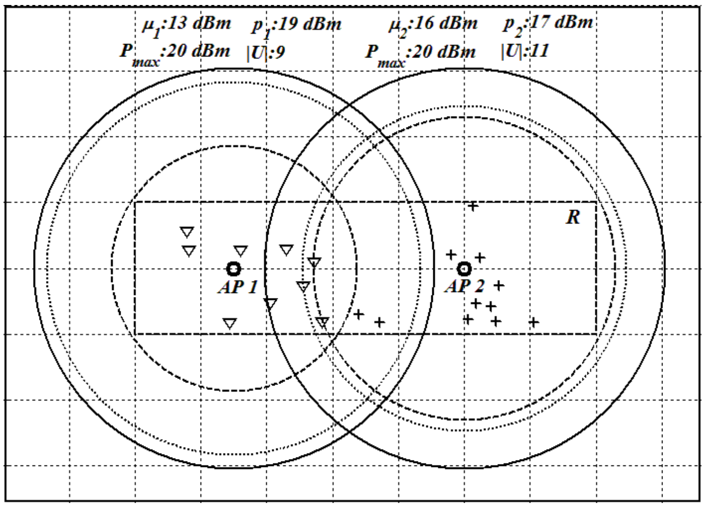

Fig. 1. A small scenario with two APs serving to a rectangular area R. Users associated with AP 1 are shown with $\bigtriangledown$ and users associated with AP 2 are shown with $+$. Innermost circles are the coverage of APs for minimum permissible power levels (μ_i). Outermost circles are the coverage of APs in the case of maximum power level. Circles between them are the actual coverage determined by the beacon power levels (p_i).

B. Service Availability Guarantee

In sparse AP deployments, reducing beacon signal level may expose some uncovered areas (gaps), which violate service availability. However, in our algorithm we propose a minimum permissible power level assignment of AP to prevent users being out of network service even if each AP set their beacon signal to the assigned minimum level. Hence, it is ensured that whole area R is covered. Let BR_i is the circular range that is served by AP$_i$. Let us denote boundaries of R and BR_i by $bd(R)$ and $bd(BR_i)$ respectively. Enclosure of R is satisfied if and only if the following conditions hold [11];

1) $bd(R)$ intersects with at least one BR.
2) If $p \in bd(R) \cap bd(BR_i)$, then $\exists j$ such that $p \in BR_j \wedge p \notin bd(BR_j)$
3) If $p \in R \wedge p \in bd(BR_i) \cap bd(BR_j)$, where $i \neq j$, then $\exists k$ such that $p \in BR_k \wedge p \neq bd(BR_k)$

Algorithm 1 GF_MMPLB

```
Input: $U$, $A$, $R$
Output: $S$
 1: $S \leftarrow \{(a, p_a = P_{max}, \mu_a = P_{max}) \mid \forall a \in A\}$
 2: $F \leftarrow NULL$
 3: while $F \neq A$ do
 4:     for $i$=1...$AP_{num}$ do
 5:         $\mu_i = \mu_i - 1$; if gap occurs $\mu_i = \mu_i + 1 \wedge F \leftarrow F \bigcup i$
 6:     end for
 7: end while
 8: $F \leftarrow NULL$
 9: while $F \neq A$ do
10:     ($S$, $d$) $\leftarrow$ Minimize_m_Coordinate($S$, $F$)
11:     $F \leftarrow F \bigcup d$
12: end while
```

Our algorithms will check these conditions for detecting possible gaps in the service area.

C. Min-Max Priority Load Balancing Problem

We considered min-max priority load balancing problem that is defined in [3]. Each AP a is given a distinct priority w_a. Priority load of an AP a, denoted by λ_a is defined as the ordered pair $\lambda_a = (l_a, w_a)$ where l_a is the aggregated load of AP a. AP a has higher load than AP b if $\lambda_a > \lambda_b$, i.e. if i) $l_a > l_b$ or ii) $l_a = l_b$ and $w_a > w_b$. A network state is said to be min-max priority load balanced if there exist no way to reduce the load of any AP without increasing the load of another AP with higher load. Min-max Priority Load Balancing algorithm proposed in [3] iteratively finds min-max priority load balanced state. However this algorithm assume that APs are deployed in such a way that there is no gap even if their beacon powers are set to the lowest level. Although they provide a small extension for scenarios where gaps are possible, they simply assume that the minimum permissible beacon powers of all APs that can guarantee full service availability are known a priori via a radio coverage survey.

Our first algorithm described in the next section is based on the Min-max Priority Load Balancing algorithm proposed in [3]. However, knowing the locations of APs, we also check the possible gaps and provide service availability as part of the algorithm. Our limitation is that there should not exist any gap when the beacon powers are set to the highest level.

III. Proposed Algorithms

A. Gap-free Min-Max Priority Load Balancing Algorithm

Gap-free Min-Max Priority Load Balancing (GF-MMPLB) algorithm is described in Algorithm 1. Algorithm uses three inputs: list of users, U, list of Access Points, A, and a rectangular closed region, R. It returns the network state S which is defined as beacon power levels, p_i, and minimum permissible power levels, μ_i of all APs in the network. Initially, for all AP i, we set p_i and μ_i to P_{max}. We decrement μ_i values of all APs one by one, until any further reduction violates the service availability in region R. This provides a gap-free environment and ensures the service availability. We have a fixed AP list called F, which is initially empty. At each iteration, a new AP is determined as fixed by the routine called Minimize-m-Coordinate, and stored in F. When all the APs are stored in F, Algorithm 1 terminates.

Routine 1 Minimize_m_Coordinate

```
Input: $S_{init}$, $F$
Output: $\tilde{S}$, $\tilde{d}$
 1: $\tilde{S} \leftarrow S_{init}$
 2: $\tilde{\Lambda} \leftarrow \Lambda$ = max$_{a \in A - F}$ $\lambda_a$
 3: $\tilde{d} \leftarrow d$ = $a$ s.t. $a \in A - F \wedge \lambda_a$ = $\Lambda$
 4: $end\_flag \leftarrow FALSE$
 5: while $end\_flag$ == $FALSE$ do
 6:     if $p_d$ == $\mu_d$ then
 7:         $end\_flag \leftarrow TRUE$
 8:     else
 9:         $p_d \leftarrow p_d$ - 1
10:         $\Lambda \leftarrow$ max$_{a \in A - F}$ $\lambda_a$
11:         $d \leftarrow a$ s.t. $a \in A - F \wedge \lambda_a$ = $\Lambda$
12:         if exist $a \in F$ s.t. $\tilde{y}_a < \lambda_a$ then
13:             $end\_flag \leftarrow TRUE$
14:         else if $\Lambda < \tilde{\Lambda}$ then
15:             $\tilde{S} \leftarrow \{(a, p_a, \mu_a)\}$
16:             $\tilde{\Lambda} \leftarrow \Lambda$
17:             $\tilde{d} \leftarrow d$
18:         end if
19:     end if
20: end while
```

Minimize-m-Coordinate routine, which is described in [3], minimizes m^th coordinate of the load vector at iteration m, without modifying the load of APs stored in F. It takes two inputs, initial network state S_{init} and F. Firstly it finds the congested AP, which is defined as the maximum loaded AP that is not in F. If it is not permissible to reduce the beacon power level of the congested AP, p_d, the routine terminates. Otherwise, it reduces p_d by one and user association is performed. If the load level of any fixed AP is increased the routine terminates. If the congestion load (i.e. load of the congested AP) of new state is less than the previous iteration, then we set new state to $\tilde{S}$, congestion load to $\tilde{\Lambda}$ and maximum load index to $\tilde{d}$. When the routine terminates, it returns last recorded $\tilde{S}$ and $\tilde{d}$.

B. Gap-free Statistical Min-Max Priority Load Balancing Algorithm

While setting minimum permissible power levels, Algorithm 1 does not consider user distribution, rather it reduces μ_i one by one. This could be a problem when APs in dense areas would not be able to reduce their beacon power levels due to minimum level limitation, although their adjacent APs may have low loads and they use high beacon power levels. Therefore we propose Gap-Free Statistical Min-Max Priority Load Balancing Algorithm (GF-SMMPLB) which categorizes APs according to their past load statistics, and consider this categorization while setting minimum permissible power levels. In other words, while GF-MMPLB reduces minimum permissible power levels of APs one by one discarding their congestion levels, GF-SMMPLB algorithm categorizes the APs into L levels considering estimated loads using the past user distribution statistics.

An AP a is in level s if the following situation holds.

$$\tilde{\lambda}_{min} + s.\Delta \leq \tilde{\lambda}_a \leq \tilde{\lambda}_{min} + (s+1).\Delta \tag{1}$$

where $\Delta = \frac{\tilde{\lambda}_{max} - \tilde{\lambda}_{min}}{L}$, $\tilde{\lambda}_a$ is the estimated load of AP a, $\tilde{\lambda}_{max}$ is the estimated load of maximum loaded AP, and $\tilde{\lambda}_{min}$

is the estimated load of minimum loaded AP. We set a level for each AP according to (1). We define A_s such that, an AP $a \in A_s$ if a is in level s.

After determining the boundaries of each level, we decrease minimum permissible power level of the APs of highest level L as much as possible. This lets the most loaded APs to decrease their beacon power levels more than the least loaded APs. Then the APs in level $L - 1$ are admitted to decrease minimum permissible power level if possible. Finally, minimum permissible power level of APs in level 1 is set. This procedure determines the minimum level of beacon coverage areas. After the assignment, GF-SMMPLB algorithm continues to execution as the GF-MMPLB algorithm, and uses the same routine Minimize-m-Coordinate. Pseudo code is given in Algorithm 2.

Algorithm 2 GF_SMMPLB

```
Input: $U$, $A$, $R$
Output: $S$
 1: $S \leftarrow \{(a, p_a = P_{max}, \mu_a = P_{max}) \mid \forall a \in A\}$
 2: $F_s \leftarrow$ NULL $\forall s \in 1...L$
 3: For each $s \in 1...L$
 4: while $F_s \neq A_s$ do
 5:     for $i$=1...$AP_{num}$ do
 6:         if $A_i \in A_s$ then
 7:             $\mu_i = \mu_i - 1$; if gap occurs $\mu_i = \mu_i + 1 \wedge F_s \leftarrow F_s \bigcup i$
 8:         end if
 9:     end for
10: end while
11: end For each
12: $F \leftarrow NULL$
13: while $F \neq A$ do
14:     ($S$, $d$) $\leftarrow$ Minimize_m_Coordinate($S$, $F$)
15:     $F \leftarrow F \bigcup d$
16: end while
```

C. Gap-free Online Min-Max Priority Load Balancing Algorithm

Although GF-SMMPLB algorithm considers past user distribution statistics while setting μ_i values, it just consider the average loads and it ignores temporal variations of user distribution. Therefore we propose another algorithm, namely Gap-free Online Min-Max Priority Load Balancing (GF-OMMPLB) algorithm which adjusts minimum permissible power levels according to current load values. In this algorithm, we first set the values of μ_i similar to GF-MMPLB algorithm. But then we permit extra reduction of μ_i values for highly loaded APs, in the expense of increasing μ_i values of their adjacent APs. This approach gives the flexibility to congestion loaded APs to decrease the beacon power level, which increases the chance of congestion load reduction. Pseudo code of GF-OMMPLB algorithm is given in Algorithm 3.

GF-OMMPLB algorithm first sets the μ_i values as in the case of GF-MMPLB algorithm. However, it uses an extended version of the Minimize-m-Coordinate routine that allows further reduction of μ_i values for highly loaded APs. This extended version reduces p_d value of congested APs as in the case of previous routine, as long as $p_d > \mu_d$. However, when p_d reduces to μ_d, it does not halt immediately, but rather it tries to further reduce the μ_d value. In order to be able to decrease μ_d value, clearly it should increase the μ_i values of its adjacent APs, but if these APs are not already in the fixed AP list. This is the job of UpdateMinPowerLevels function, which increases μ_i levels of all adjacent APs (that are not in the fixed AP list) to P_{max}, and then reduce them as much as possible. If this function could not find any feasible assignment of μ_i values that satisfy full coverage, it returns NULL and the routine halts. Otherwise, it returns $\tilde{M}$, updated μ_i values of all APs. Since minimum permissible power of the congested AP is now reduced, it continues with reducing its beacon power level (p_d), and check whether congestion load is decreased without increasing loads of fixed APs, as in the case of Minimize-m-Coordinate routine.

Algorithm 3 GF_OMMPLB

```
Input: $U$, $A$, $R$
Output: $S$
 1: $S \leftarrow \{(a, p_a = P_{max}, \mu_a = P_{max}) \mid \forall a \in A\}$
 2: $F \leftarrow NULL$
 3: while $F \neq A$ do
 4:     for $i$=1...$AP_{num}$ do
 5:         $\mu_i = \mu_i - 1$; if gap occurs $\mu_i = \mu_i + 1 \wedge F \leftarrow F \bigcup i$
 6:     end for
 7: end while
 8: $F \leftarrow NULL$
 9: while $F \neq A$ do
10:     ($S$, $d$) $\leftarrow$ Minimize_m_Coordinate_Extended($S$, $F$, $R$)
11:     $F \leftarrow F \bigcup d$
12: end while
```

Routine 2 Minimize_m_Coordinate_Extended

```
Input: $S_{init}$, $F$, $R$
Output: $\tilde{S}$, $\tilde{d}$
 1: $\tilde{S} \leftarrow S_{init}$
 2: $\tilde{\Lambda} \leftarrow \Lambda = \max_{a \in A - F} \lambda_a$
 3: $\tilde{d} \leftarrow d = a$ s.t. $a \in A - F \wedge \lambda_a = \Lambda$
 4: $end\_flag \leftarrow FALSE$
 5: while $end\_flag == FALSE$ do
 6:     if $p_d == \mu_d$ then
 7:         if $\mu_d == P_{min}$ then
 8:             $end\_flag \leftarrow TRUE$
 9:         else
10:             $\mu_d \leftarrow \mu_d$ - 1
11:             $M \leftarrow \{\mu_a | \forall a \in A\}$
12:             $\tilde{M} \leftarrow$ UpdateMinPowerLevels($APLoc$,$M$,$R$,$F \bigcup \{d\}$)
13:             if $\tilde{M}$ == $NULL$ // if gap exists then
14:                 $\mu_d \leftarrow \mu_d$ + 1 // undo $\mu_d$ reduction
15:                 $end\_flag \leftarrow$ TRUE
16:             else
17:                 $\{\mu_a\} \leftarrow \tilde{M}$
18:                 $p_d \leftarrow p_d$ - 1
19:                 $\Lambda \leftarrow \max_{a \in A - F} \lambda_a$
20:                 $d \leftarrow a$ s.t. $a \in A - F \wedge \lambda_a = \Lambda$
21:                 if exist $a \in F$ s.t. $\tilde{y}_a < \lambda_a$ then
22:                     $end\_flag \leftarrow TRUE$
23:                 else if $\Lambda < \tilde{\Lambda}$ then
24:                     $\tilde{S} \leftarrow \{(a, p_a, \mu_a)\}$
25:                     $\tilde{\Lambda} \leftarrow \Lambda$
26:                     $\tilde{d} \leftarrow d$
27:                 end if
28:             end if
29:         end if
30:     end if
31: end while
```

IV. Simulation Results

We perform extensive set of simulations to provide comparison of the algorithms stated above in terms of data load of APs and bandwidth rate of users. We also compared our algorithms with Shortest Signal First (SSF), which is an

TABLE II. BITRATE VS SNR IN 802.11G

SNR $\geq$ 1dB $\rightarrow$ 1 Mbps
SNR $\geq$ 3dB $\rightarrow$ 2 Mbps
SNR $\geq$ 5dB $\rightarrow$ 5 Mbps
SNR $\geq$ 6dB $\rightarrow$ 9 Mbps
SNR $\geq$ 7dB $\rightarrow$ 12 Mbps
SNR $\geq$ 9dB $\rightarrow$ 18 Mbps
SNR $\geq$ 13dB $\rightarrow$ 24 Mbps
SNR $\geq$ 17dB $\rightarrow$ 36 Mbps
SNR $\geq$ 20dB $\rightarrow$ 48 Mbps
SNR $\geq$ 22dB $\rightarrow$ 54 Mbps

association technique such that users are connected to the AP that has strongest signal transmission as stated in IEEE 802.11 standard.

In our simulation, we assumed a wireless local area network with 12 APs equipped with 802.11g technology is deployed in a 640x480 m^2 region R where the APs positions are equidistant to their neighbors. The distance between APs is set to 160 meters and AP bandwidth is set to 10 Mbps. In region R, it is assumed that APs provide a full network coverage by transmitting with highest power level. User-AP association is determined by the users according to the Signal-to-Noise ratio (SNR) of the beacon signals sent by the APs. We also consider rate adaptation employed by 802.11g, so that SNR values determines the bit rate between user and AP. We used SNR-to-bit rate calculation stated in [10], as indicated in Table II.

Additionally, P_{max} and P_{min} are given as 20 dBm and 10 dBm respectively. To simulate the indoor environment, we chose the path loss exponent of 3.3 [7], so that the path loss formulation is given as $PL(d) = 40 - 10 \times 3.3 \times \log d$, where d is the distance between user and AP. The background noise is set to -93dBm. Using this model, the minimal coverage range is found as 75 m and maximal coverage range is found as 150 m. The network setup can cause coverage holes between APs, however the algorithms prevent occurrence of this problem.

A user load on an AP can be calculated using the bit rate information. We consider the system performance metric described in [2] and define the load of an AP_{i} as follows,

$$\tilde{\lambda}_{\text{i}} = \sum_{u \in U_{\text{i}}} \frac{1}{r_{u,\,i}} \tag{2}$$

where $r_{\text{u,i}}$ is the bit rate with which the user u communicates with AP_{i} and U_{i} is the set of users that are associated with AP_{i}.

Moreover, while simulating GF-SMMPLB algorithm, we choose L=3 and we assumed that distribution of the users are time-invariant, hence loads can be perfectly estimated.

Figure 2 depicts the load of APs in a network environment with 300 randomly distributed users in region R. The x-axis represents the AP's index and the y-axis represents average load value of an AP. At each simulation run, the APs are sorted by the load values in decreasing order. After 50 simulation runs, the λ_{i} of each AP n is computed by finding the average value of the n^{th} highest loaded AP. The load values at each AP index are joined to make the graph sightly. In Figure 2, GF-OMMPLB algorithm is shown with a thick solid line, GF-SMMPLB is plotted with a thin solid line and GF-MMPLB algorithm is represented with a thick dashed line. Although GF-MMPLB algorithm generates a load vector with fairly lower

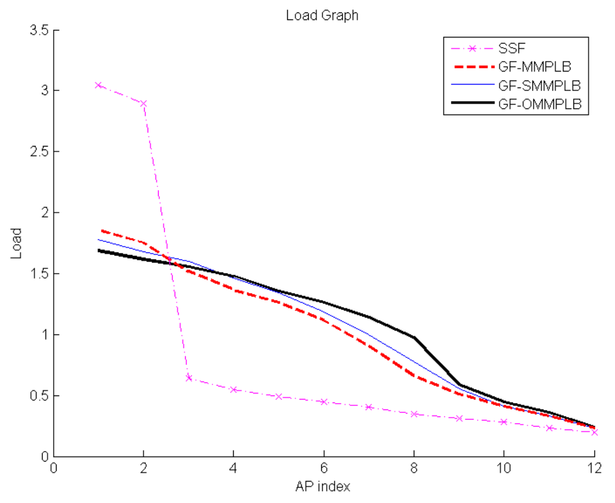

Fig. 2. Load comparison of algorithms with 300 randomly distributed users

TABLE III. LOADS OF MOST CONGESTED TWO APS IN FIGURE 2

AP Index	SSF	GF-MMPLB	GF-SMMPLB	GF-OMMPLB
1	3.0467	1.8585	1.7803	1.6868
2	2.8925	1.7534	1.6803	1.6148

lexicographical order, GF-OMMPLB algorithm decreases the maximum load value of APs better than other algorithms. All proposed algorithms clearly outperform the SSF algorithm. The loads of most congested two APs are shown in Table III.

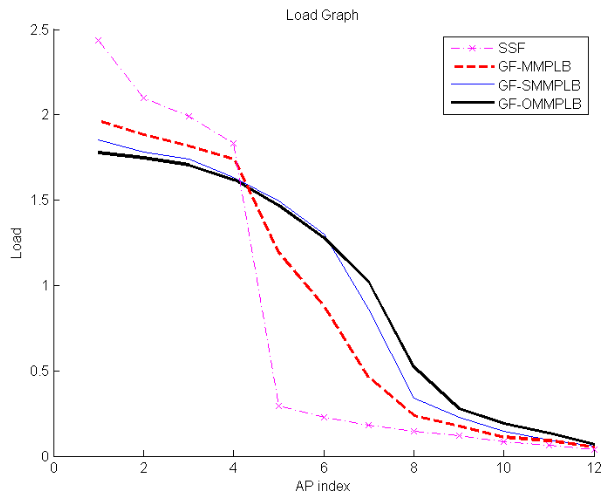

Fig. 3. Load comparison of algorithms with 300 random users where 4x50 users are concentrated on four APs

Figure 3 presents a network environment with unbalanced distribution of 300 users. In this setting, we determine four hot-spot regions and we distribute 50 users to each of these regions. Last 100 users are distributed randomly in R. Each hot-spot is square shaped of 150x150 m^2. Existence of hot-spots causes different levels for APs, so that statistical algorithm uses the level information to determine the μ_{i} value of APs. Hence, for this setting, GF-SMMPLB performs better than GF-MMPLB algorithm and decreases the load in the hot-spot

TABLE IV. LOADS OF MOST CONGESTED FOUR APS IN FIGURE 3

AP Index	SSF	GF-MMPLB	GF-SMMPLB	GF-OMMPLB
1	2.4375	1.9632	1.8530	1.7775
2	2.0968	1.8833	1.7822	1.7472
3	1.9919	1.8197	1.7414	1.7058
4	1.8331	1.7412	1.6317	1.6194

areas by approximately 6%. On the other hand, GF-OMMPLB algorithm decreases the load of congested APs even better, and provides up to 10% load reduction in hot-spot areas, compared to GF-MMPLB algorithm. SSF algorithm performs ineffective for the first four APs, because the hot-spots create immense load at the four APs. The loads of most congested four APs are shown in Table IV.

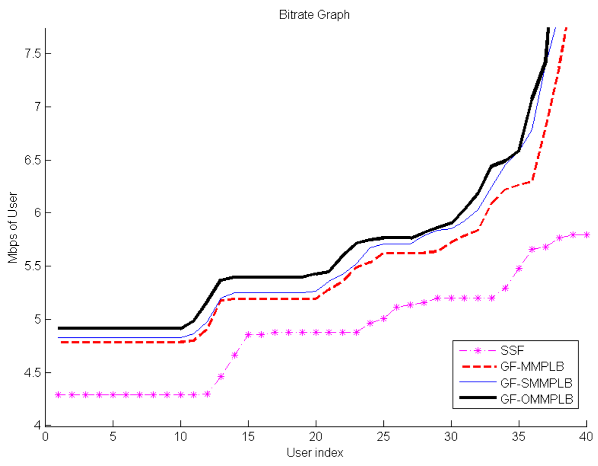

Fig. 4. Per-user bandwidth rate for 40 users that experience lowest rates in hot-spot scenario

Bandwidth distribution among the first 40 users that experience lowest rates is depicted in Figure 4, where the simulation settings are set to same configuration with the previous experiment. It is observed that GF-OMMPLB algorithm provides better per-user bandwidth for the users in crowded areas, because the load balancing among the APs is done more effectively. So we can say that GF-OMMPLB algorithm performs best in terms of providing fair services to users. GF-SMMPLB also increases per user bandwidth in hot-spot areas and it is more fair compared to GF-MMPLB algorithm.

V. CONCLUSION

Cell-Breathing technique is a promising approach for load balancing in IEEE 802.11 WLAN networks, since it does not require any modifications in the user equipment. Nevertheless, it can suffer from AP service availability problem since coverage reduction can create some unserviced areas. In this paper, we provided a min-max priority load balancing algorithm which sets a minimum permissible level for the beacon power of each AP such that no gap occurs. Additionally, we proposed two novel enhancements to this algorithm. Firstly, we described an algorithm that utilizes the statistical past user distribution information while deciding on the minimum permissible beacon power levels. Secondly, we provided an online algorithm which adjusts the minimum permissible power levels according to current load values. Simulations show the fairness of the proposed load balancing algorithms. As a future work, we are planning to extend our work for the case of mobile users.

ACKNOWLEDGMENT

This work is supported by Marmara University BAPKO under grant number FEN-D-121114-0370.

REFERENCES

[1] P. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in wireless lans: Algorithms and evaluation," *IEEE Transactions on Mobile Computing*, vol. 6, no. 2, pp. 164–178, 2007.

[2] Y. Bejerano, S.-J. Han, and L. Li, "Fairness and load balancing in wireless lans using association control," in *Proc. ACM MobiCom*, 2004, pp. 315–329.

[3] Y. Bejerano and S. jae Han, "Cell breathing techniques for load balancing in wireless lans," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 735–749, June 2009.

[4] P. Calhoun, "Control and provisioning of wireless access points (capwap) protocol binding for ieee 802.11, rfc 5416," in *IETF Standard*, 2009.

[5] H. Gong, K. Nahm, and J. Kim, "Distributed fair access point selection for multi-rate ieee 802.11 wlans." in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, 2008.

[6] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection. in proceedings of the 4th international conference on mobile systems, applications and services," in *Proc. ACM MobiSys*, 2006.

[7] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[8] V. S., K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley, "Facilitating access point selection in ieee 802.11 wireless networks," in *Proc. ACM SIGCOMM conference on Internet Measurement*, 2005.

[9] D. Schwab and R. Bunt, "Characterising the use of a campus wireless network," in *Proc. IEEE INFOCOM*, vol. 2, March 2004, pp. 862–870 vol.2.

[10] W. Stallings, *Wireless Communications & Networks (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.

[11] S. Wang, J. Huang, X. Cheng, and B. Chen, "Coverage adjustment for load balancing with an ap service availability guarantee in wlans," *Wireless Networks*, vol. 20, no. 3, pp. 475–491, 2014.

[12] L. H. Yen and T. T. Yeh, "Snmp-based approach to load distribution in ieee 802.11 networks," in *Proc. IEEE VTC*, 2006.

[13] Y. Yukuda and Y. Oie, "Decentralized access point selection architecture for wireless lans deployability and robustness," in *Proc. IEEE VTC*, 2004.