2013 IEEE 33rd International Conference on Distributed Computing Systems

S^3: Characterizing Sociality for User-Friendly Steady Load Balancing in Enterprise WLANs

Chaoqun Yue, Guangtao Xue, Hongzi Zhu, Jiadi Yu, Minglu Li
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Email: xue-gt@cs.sjtu.edu.com

***Abstract*—Traffic load is often unevenly distributed among the access points (APs) in enterprise WLANs. Such load imbalance results in sub-optimal network throughput and unfair bandwidth allocation among users. In this paper, we collect real traces from over twelve thousand WiFi users in Shanghai Jiao Tong University. Through intensive data analysis, we find that user behavior like leaving together may cause significant AP load imbalance problem. We also observe from the trace that users with similar application usage have the potential to leave together. Inspired by those observations, we propose an innovative scheme, *Social-aware AP Selection Scheme*(S^3), which can actively learn the sociality information among users trained with their history application profiles and elegantly assign users based on the obtained knowledge. Both real prototype implementation and simulation results show that S^3 is feasible and can achieve 41.2% balancing performance gain on average.**

***Index Terms*—IEEE 802.11, enterprise WLANs, load balancing, social behavior based AP selection**

I. INTRODUCTION

In recent years, IEEE 802.11 wireless LANs (WLANs) have been widely deployed in enterprises, public areas and homes. Studies [13] [2] on operational WLANs have shown that the traffic load is often unevenly distributed among the access points (APs). In enterprise WLANs, each user scans all available APs and associates itself with the AP that has the strongest received signal strength indicator (RSSI) by default, ignoring the load condition on this AP. It is often the case that APs suffer from severe load unbalancing, which hampers the network from providing maximum throughput and fair services to its users.

To solve the load balancing problem, however, is very difficult. The reason is two-fold. First, without knowing future traffic demands on individual APs, it is very hard, if not impossible, to make an optimal assignment or adjustment of WLAN users among a set of APs. Second, it is inevitable to cause link disruptions when dynamically migrating users from heavy-load APs to those light-load ones. Although it is possible for a user to maintain multiple links at the same time, it requires all users to have extra hardware and therefore is infeasible in practice.

Although the AP load balancing problem has been studied for years, it still has not been thoroughly solved. In the literature, the existing schemes can be classified into two categories. One is user-arrival-based methods [14] [15] [18], where the AP with the least workload will be chosen to serve a new coming user. Such schemes can adjust the load balancing only when there are new users joining but incapable in improving load balance when network traffic churns happen (e.g., joining and leaving of users and changes of running applications). Therefore, these schemes sacrifice load balancing performance for excellent user experience. In contrast, the other one is online load balancing schemes which can rapidly adjust the traffic load among APs. When traffic churns are highly dynamic, those schemes can achieve good load balancing performance but would also cause unpleasant constant connection disruptions. As a result, there is no existing scheme, to the best of our knowledge, to successfully tackling the load balancing problem in enterprise WLANs that can achieve superior load balancing while still preserving good user experience.

In this paper, we take an empirical methodology to study the load balancing problem in enterprise WLANs. We have collected real WLAN trace from Shanghai Jiao Tong University involving more than 12,000 users over three months from July to October, 2012. After intensively mining and analyzing the trace with regard to the load balancing problem, we first find that the state-of-art strategy adopted in enterprise WLANs can hardly achieve load balance. For instance, we find that for about 20% peak-hour time and 60% off-peak-hour time traffic load on APs is rather unbalanced in our trace using the Least Loaded First (LLF) scheme [9]. We further seek for the fundamental leading factors and find that the churns of WLAN users have the most significance in causing the high dynamics of traffic load on APs. Moreover, we observe obvious social characteristics of users behavior, namely, coming to or leaving the network together (called *co-coming* or *co-leaving* events). We dig the trace and find that co-leaving users have very similar application profiles. With this insight, we propose an innovative scheme, *S*ociality-aware AP *S*election *S*cheme or S^3, for user-friendly load balancing in enterprise WLANs. The core idea of S^3 is to characterize the sociality of users by grouping users with similar application profiles. With the knowledge of application profiles and social relationships of users, S^3 elegantly distributes co-leaving users to a set of APs considering the current work load on those APs as well. The main advantage of S^3 lies in two folds. First, S^3 is user-friendly as it does not migrate users from an AP to another.

1063-6927/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDCS.2013.56

Second, it is very resilient to network churns as it can resist sudden traffic demand changes caused by co-leavings. The real prototype implementation verifies the feasibility of the S^3 design. Through extensive accurate trace-driven simulations, we demonstrate the efficacy of S^3 and the results show that S^3 can achieve 41.2% balancing performance gain on average comparing with LLF.

The main contributions of this paper are highlighted as follows:

- We have collected tremendous amount of WLAN usage trace containing entire network users of over twelve thousand. By data mining, we find the state-of-art strategy adopted in enterprise WLANs can hardly achieve steady yet user-friendly load balance and the leading factor is the churns of network users.
- We have also observed that strong social behavior among WLAN users, which can be characterized using application profiles of users. We propose a user-friendly and steady AP selection scheme S^3 leveraging all these key observations.
- We have implemented a small-scale prototype to verify the feasibility of S^3 and conducted extensive trace-driven simulations to evaluate its performance. Results have demonstrated the effectiveness of S^3.

The remainder of this paper is organized as follows. We presents the related work in Section II. In Section III, we describe the details of our collected data set and exhibit the social relations through the analysis of the real trace and then study the impacts of the unbalanced user leaving on the AP performance. We characterize the relation between users' application profile and their social behaviour. In the fourth section, we provide the system model and propose the algorithm. We describe the prototype in Section V. Section VI describes the methodology to evaluate the performance of our AP selection algorithm and presents the results. Finally, we present concluding remarks and outline in Section. VII.

II. RELATED WORKS

There exists many works on AP load balancing. The existing algorithms can be divided into the following two classes:

One attempt to adjust the users during the runtime [12], i.e., moving users from one AP to another. These approaches have been widely challenged due to the large computational overhead. In addition, users will experience disruptions during the immigration, which is undesired or even unacceptable.

Another majority of approaches distribute user when they arrive. Nicholson, A.J. et al [15], [14] design the AP selection algorithm based on the assessment of the AP. In its design, a node tentatively associates to each of the AP and evaluates the bandwidth and delay of the AP towards the Internet. The AP with a better service quality will be selected. In the work [13], an AP-selection strategy is proposed that takes two major factors into account. One is the individual user throughput, and the other is the impact on the performance of other users who are already associated with the AP. In [17], a distributed load balancing system is introduced which takes into account the AP loads defined as the aggregated uplink and downlink traffic through the AP. The author in the paper [19] proposes an online AP association strategy that maximizes the minimum throughput among all users at the cost of an acceptable overhead. A distributed algorithm [20] is proposed where the APs can tune their cell size according to their load and their neighbors loads in a way transparent to the end users and thus improve the fairness and performance. The author in [18] proposes to balance the load on the basis of the entire network. In the design, demand clusters will select the AP that is able to provide ample bandwidth, rather than the closest AP which often has the largest signal level. In [21], the authors proposed a unique solution called SAP (Smart Access Point), which smartly balancing the network load across multiple interfaces based on users time-varying traffic load conditions. All these approaches, however, only consider the arrival of the users, and none of them have taken the social relations between users into account.

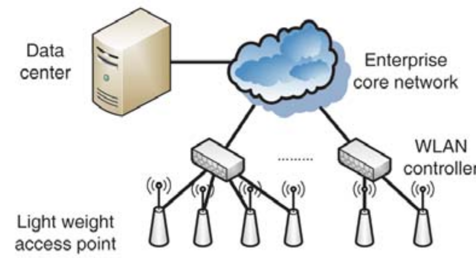

Fig. 1. Structure of the WLAN in SJTU

The other category of related work studies the social relations in users. Hsu, W. et al [23] studied the user behaviors in AP access and explicit patterns are observed. In the work [24], the authors tried to understand the inter-user interaction in wireless environments by investigating the inter-node encounters. The Small World concept is explored. Furthermore, the author displays the feasibility of an infrastructure-less network, where most of the nodes can be reached through utilizing inter-node connectivity and encounters. These works only consider the social relations among users and never study the impact of such relations to the AP selection and balance in WLANs.

III. MEASUREMENT ANALYSIS

In this section, we first introduce the trace data that we have collected and then examine the AP load unbalancing problem with the trace. Next, we analyze the major factors that cause the load unbalancing problem. With the observation that churns of users play an essential role, we finally reveal the sociality of users that triggers the churn.

A. Collecting Empirical Trace Data

We collect WLAN usage trace data from SJTU, a prestigious university in mainland of China. Fig. 1 illustrates a typical enterprise WLAN deployed in SJTU, which consists of three major entities, i.e., light-weighted APs, WLAN controllers and a back-end data center. A WLAN controller taking the charge of several APs in vicinity is responsible for

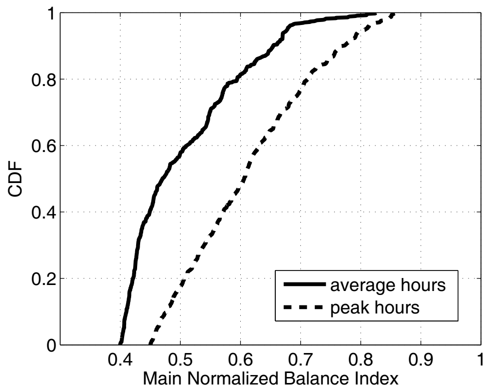

Fig. 2. CDF of the normalized balance index over all controllers

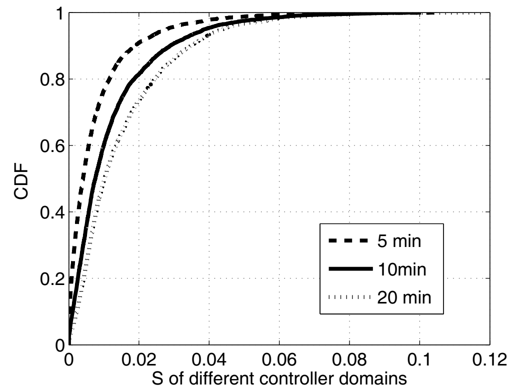

Fig. 3. CDF of the variance of balance index over all controllers

assigning users to specific APs within its domain. The state-of-art strategy adopted by a controller is to assign a new user to the AP with the least traffic load (or with the least number of users). We collect trace from the back-end data center, which records all login information.

For the study in this paper, we have collected trace data over three months from July to October in the year of 2012 which involve 12,374 users collected from 334 APs deployed in 22 buildings. The specific fields in logged records include: user identifiers (i.e., MAC addresses of wireless cards), connected time stamps (the time instance when a user successfully connected to an AP), disconnected time stamps (the time instance when a user disconnected from an AP), accessed AP, and the served traffic amount (the total traffic amount a user sent to or received from an AP during a connection). In addition, from the core network routers, we also obtain all WLAN traffic information including: the source and destination IP addresses of a packet, transportation layer and application layer ports (e.g., tcp, HTTP, DNS, SIP). By analyzing the port combination using certain heuristics [1], concrete applications can be accurately identified. In our trace, all user identifiers are processed with hash functions (e.g., SHA) to remove privacy concerns. For proprietary reasons, the results presented in this paper are normalized, which, however, does not change the range of the metrics used in this study. Furthermore, the missing information due to normalization does not affect the understanding of our analysis.

As there are a vast number of applications involved in the trace, we examine the top 30 in terms of generated traffic volume, which constitute the vast majority of all data traffic. Thus, understanding the remainder is not critical for the purpose of network engineering [2]. Furthermore, we categorize these top applications into the following six application realms: 1) IM, 2) P2P, 3) music, 4) E-mail, 5) video, and 6) web-browsing.

B. Load Balancing Problem in Enterprise WLANs

As load balancing is of great importance to the network performance and user experience, we now examine the load balancing problem with our trace. To better quantify the load balance level among a set of APs, we use the following balancing index definition [26]:

Definition : Given n APs, let T_i denote the throughput of the i-th AP, $i = 1, ..., n$, the balancing index is defined as:

$$\beta = \frac{(\Sigma T_i)^2}{n * \Sigma T_i^2}.$$

We further define the normalized balancing index as:

$$\overline{\beta} = \frac{\beta - \frac{1}{n}}{1 - \frac{1}{n}}$$

This index has been widely used in the literature to assess the load balancing performance. Other fairness metrics, such as *max-min* [16] and *proportional fairness* [11], may also be used. With different load-balancing strategies, the balancing index ranges from $\frac{1}{n}$ to 1 with larger index value indicating better balancing level. Fig. 2 shows the cumulative distribution function (CDF) of normalized balance index calculated between all APs under a WLAN controller over all controllers with the trace. Peak hours in the figure refer to the hours from 10:00 to 11:00 and from 15:00 to 16:00, when the network throughput reaches the peaks during a day. It can be seen that about 20 % of time during peak hours and about 60% of time during a workday, the traffic load on APs is rather unbalanced (balance index is less than 0.5). This indicates that enterprise WLANs cannot achieve good load balancing performance using the state-of-art AP selection strategy which is LLF [9].

C. Leading Factors Analysis

There are two cases where the original AP load balance may be broken. One is when users suddenly change their running applications, which may incur sudden changes of traffic demands. The other is when the number of users on APs suddenly drops unevenly, which can also result in sudden changes of traffic demands. We first look at each case in this subsection.

1) Factor of Application Dynamics: To know whether the change of applications accounts, we analyze the variations of balance index caused by application dynamics. Specifically, we divide the trace with time periods of an hour and remove the traffic amount generated by users who just came or left during a time period. For each time period, we further divide the time period into n sub-time-periods. We calculate the balance indexes of all APs under a controller in the ith sub-time-period, denoted as β_i, and calculate the variance of $\beta_i, i = 1, \cdots, n$. We defined the variance of balance index S as:

$$S_i = \frac{\beta_i - \beta_{i-1}}{\beta_{i-1}}$$

Fig. 3 plots the CDFs of the S over all time periods and all controllers with the length of a sub-time-period equal to five minutes, ten minutes and twenty minutes, respectively. It can be seen that more than 80% of variance is less than 0.02 with ten-minute sub-time-periods. This result shows that the balance index does not change suddenly with fixed users.

2) Factor of User Dynamics: We now investigate the impact of churns of users on the load balancing problem of WLANs. We use a similar method to quantify balance index of the number of users β_{num} among all APs under a WLAN controller. Fig. 4 shows an example of the relationship between the balance index of the number of users β_{num} and the balance index of traffic load $\beta_{traffic}$ on an AP during a workday from 8:00 to 24:00. It can be seen that two plots are very similar in layout. Particularly, when β_{num} drops, $\beta_{traffic}$ also drops (indicated by the dotted lines in the figure). Note that the number of users associated with an AP is affected by both joining and leaving of users. With the state-of-art arrival-based algorithm LLF where a new coming user is allocated to the AP with the least workload, therefore, joining users will be well tended by the LLF scheme and will not cause a serious load unbalance. However, when multiple users leave an AP in a short period of time, the traffic demands on this AP will dramatically drop and may lead to load unbalance. In this case, LLF can not recover from such load unbalance. Therefore, we have the conclusion that the churn of users, especially, co-leavings of users, plays an essential role in causing load unbalance in enterprise WLAN.

D. Revealing User Sociality behind the Churns

1) Sociality of WLAN Users: in this section, we investigate the user activities which might cause unbalanced leavings.

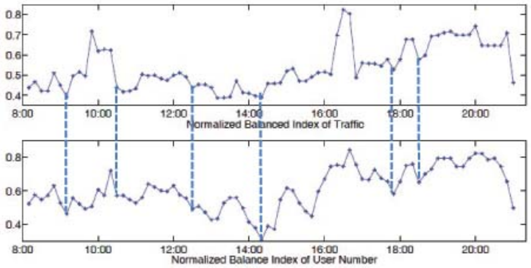

Fig. 4. An example of the relationship between the balance index of the number of users and the balance index of traffic load

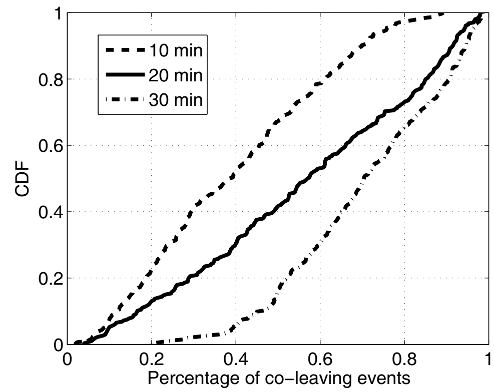

Fig. 5. CDF of the number of co-leaving events to the total number of leaving events over all users

For example, people in an enterprise domain often have routine activities, such as classes in schools and department meetings in corporations. These social activities may have great influence on the way people use WLAN. From the perspective of AP- accessing behavior, we study two main events in the trace data that may reflect those social activities as follows:

- *Encountering* is referred to as a pair of users keep the connections with the same AP for a certain period of time. Notice that a co-coming is not necessary to lead to an encountering as one of two users may leave sooner than the given period of time.
- *Co-leaving* is referred to as a pair of users leave the same AP at the same time or within a short period of time.

Two users are said to have a social relationship if they share common aforementioned events. Indeed, it is likely for two users to have common events by chance instead of attending the same social activities, especially when a relatively long time period of time is used to extract co-leaving events. Such fake social relationships are random and have no capability to predict future AP access behavior of users. We take fake social relationships as noise and diminish its effect by carefully choosing appropriate time periods for event extraction and aggregating multiple common events between the same pair of users for a single social relationship.

In order to investigate the probability of co-leaving events, we plot the CDF of the number of co-leaving events to the total number of leaving events over all users in Fig. 5, using three different time periods of time for event extraction, namely, ten minute, twenty minutes and half an hour. It indicates that most users show strong sociality in their AP access behavior and do not leave an AP independently.

2) Capturing User Sociality: As different users have different application usage profiles, we used normalized history traffic volumes of the six major application categories mentioned before to characterize the application interest of a user. To answer the question that how much history data are sufficient to accurately capture the user interest, we examine the temporal correlation of user application profiles. More

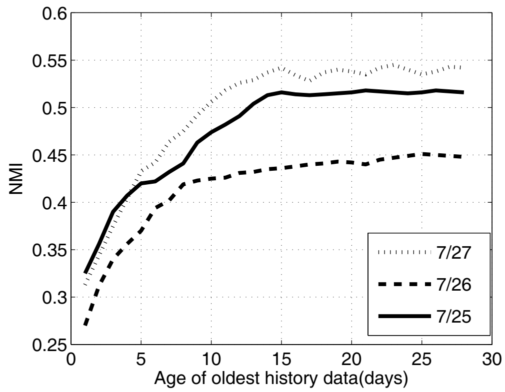

Fig. 6. Average NMI vs. n between point and cumulative traffic vectors.

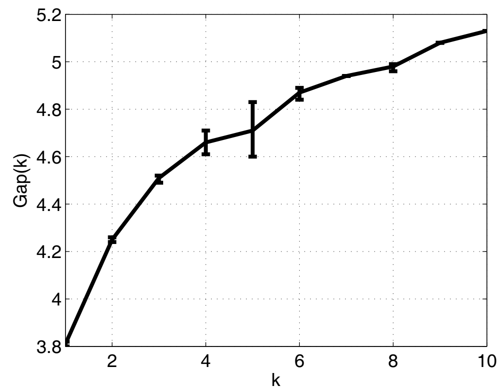

Fig. 7. Gap statistic for varying values of k

specifically, for a particular user u, we consider the application profile $T_x(u) = (a_u^0, a_u^1, ..., a_u^6)$ where a_u^i is the volume of traffic caused by applications in the ith category $(1 \leq i \leq 6)$ associated with user u on day x. We also have the vector for user u from day $x-n$, $T_{x-n}(u)$. We can then compute the entropy of the joint distribution of $T_x(u)$ and $T_{x-n}(u)$ over applications 1 through 6. The mutual information of $T_x(u)$ and $T_{x-n}(u)$, $I(T_x(u, T_{x-n}(u))$, can be obtained via the joint entropy $H(T_x(u), T_{x-n}(u))$ and the entropies of $T_x(u)$ and $T_{x-n}(u)$ as follows:

$$\begin{aligned} I(T_x(u), T_{x-n}(u)) = H(T_x(u)) \quad &+ \quad H(T_{x-n}(u)) \\ &- \quad H(T_x(u), T_{x-n}(u)) \end{aligned}$$

where H(.) is the entropy. We define the Normalized Mutual Information (NMI) by

$$NMI(T_x(u), T_{x-n}(u)) = \frac{I(T_x(u), T_{x-n}(u))}{H(T_x(u))}.$$

We examine the change in the NMI when we make more historical data available. Instead of using solely the data from day $x-n$, we aggregate the data from day $x-1$ through day $x-n$. The resulting traffic vector is thus $\sum_{i=1}^{n} T_x(u-i)$ Then, we consider the mutual information between $T_x(u)$ and $\sum_{i=1}^{n} T_x(u-i)$ normalized by $H(T_x(u))$. We show the results in Fig. 6. It is quite clear to see that the NMI increases until about $n = 15$, when it hits a plateau and stabilizes. This means that adding more history data before 15 days does not help (but does not hurt either) in building application profiles.

By further checking with those co-leaving users, we find that users who having higher probability to leave together also have more similar application profiles. Inspired by this connection, we further investigate whether two users who share similar application usage profiles would often leave together. For this purpose, we cluster users using their normalized traffic volumes of applications. We utilize a well-known unsupervised clustering algorithm called k-means to cluster application distributions of cells. k-means algorithm is a simple yet effective technique to cluster feature vectors into a predefined k number of groups [4]. The selection of appropriate value of k is crucial and is an open research problem [5]. Several heuristics have been proposed in prior literature, which primarily focus on the change in intra-cluster dissimilarity for increasing values of k [6], [7]. One of the most well-known heuristic, called gap statistic, involves comparing the change in intra-cluster dissimilarity W_k for given data and that for a reference null distribution [8]. Gap statistic provides a statistical method to find the elbow of intra-cluster dissimilarity W_k as the values of k varies. Gap statistic is defined as:

$$Gap(k) = \frac{1}{B} \sum_{b=1}^{B} log(W_{kb}) - log(W_k)$$

where W_{kb} denotes the intra-cluster dispersion of a reference data set from a uniform distribution over the range of the observed data. Using gap statistic, the optimal value of k can be chosen to be the smallest one for which:

$$Gap(k) \geq Gap(k+1) - \sigma_{k+1}$$

where σ denotes the standard deviation of intra-cluster dispersions in reference data sets. Fig. 7 shows the plot of gap statistic for varying values of k. We observe that $Gap(4) \geq Gap(5) - \sigma 5$, so we select the optimal value of $k = 4$. After selecting the value of $k = 4$ using gap statistic, we apply the k-means clustering algorithm to cluster application usage patterns of users into four groups.

To get a clue about the optimal clustering result, we plot cluster centroid of four user groups as Fig. 8 shows. We observe that a user can be divided into a distinct group

TABLE I
POSSIBILITY OF LEAVING TOGETHER BETWEEN DIFFERENT USAGE TYPES

τ	*type1*	*type2*	*type3*	*type4*
type1	0.51	0.23	0.31	0.17
type2	0.23	0.66	0.31	0.26
type3	0.31	0.31	0.54	0.22
type4	0.17	0.26	0.22	0.61

according to its application usage profile. We label these four groups using *type1*, *type2*, *type3*, and *type4*. Let $\tau(type_i, type_j)$ represents the mean possibility that a pair of tags from group $type_i$ and $type_j$ will leave together.

Table 1 shows that a user is more likely to leave together with another user in the same group than other users (this can be seen that $\tau(type_i, type_j)$ has greater values in the diagonal line of the table). We consider to utilize this strong pattern for forecasting the co-leaving events.

E. Summary

In summary, we have the following key observations: 1) the throughput of an AP is tightly connected with the number of users associated with that AP; 2) the churns of WLAN users especially caused by co-leaving events are the key factor to load unbalance status of APs; 3) user application usage profiles can be used to predict social behavior of users such as co-leavings.

IV. Social-aware AP Selection Algorithm Design

In this section, we first introduce the social relation index, a key metric to quantify social relationships between users. We then formally define the AP selection problem. At last, we present our social-aware AP selection algorithm for enterprise WLANs.

Given two users u and v, the social index of them is defined as :

$$\delta(u,v) = P(L(u,v)|E(u,v)) + \alpha * \tau(U,V)$$

where $L(u,v)$ and $E(u,v)$ denote the co-leaving and encountering events between u and v, respectively. In other words, $(L(u,v)|E(u,v))$ is the conditional probability that u and v encounter at the same AP and then leave the AP in unison. However, if the pair of users have not encountered each other before, we need other information to guess the possibility that they will leave together. $\tau(U,V)$ is mentioned in the last section where $u \in type_U, v \in type_V$ and α is an constant coefficient. Thus, a high social relation index implies a stronger relation between users and vice verse. We expect the social relation index can effectively forecast the co-leaving events between users which affect the balanced index in the same controller domain.

A. Problem Statement

The original AP balancing problem is to distribute the users to different APs so that all the APs are kept balanced in all the time, i.e., min($\Sigma\beta$). In practice, this problem has no optimal solution because the optimal solution requires the exact leaving time for each user. Such information is about the future and can never be obtained. Fortunately, we recall that the main cause of the AP unbalance is the user co-leaving events. Alternatively, we take another approach towards the optimal AP balance scheme. We try to distribute user pairs of tighter social relations to different APs so that users in the same APs are diversified. In other words, they have fewer social relations and will be unlikely to present similar access behaviors. The problem is formally defined as follows.

Definition 1: Given a user willing to subscribe to an AP and there are N APs can be accessed. suppose this user u has a demanded throughput w(u) and there is a social relation index between any pair of users $(u,v), u,v \in \sum_{i=1}^{n} AP_i$. Assume the bandwidth of the APs are W(i), i = [1, m], the problem is to find an allocation for the user to an AP such that

Object:

$$1\ )\quad min : \sum_{i=1}^{N} \sum_{\forall u,v \in AP_i} \delta(u,v)$$

$$2\ )\quad min(\beta_{new})$$

subject:

$$\sum_{u \in a} w(u) < W(i), i \in [1,n]$$

Here the constraint $\sum_{u \in a} w(u) < W(i), i \in [1,n]$ is due to the fact that the aggregated throughput demands cannot exceed the provided AP bandwidth.

We have the following theorem:

Theorem 1: The load balancing problem defined in Definition 1 is $NP-complete$.

The proof of Theorem 1 can be conducted by inducing the weighted maximum cut problem, a well known NPC problem to the load balancing problem. Due to the page limitation, we omit the detail proof.

The problem proposed here is also a multi-objective optimization problem. To achieve the solution, we consider Object 1) as the main object to achieve. As our scheme let balance index not be too bad because of the erase of the co-leaving events, we just need to prevent the balance index from decreasing too much which may be caused by the user distribution.

Assume that each user to be assigned is a vertex in an undirected graph. We define an edge between two users if the social relation index between this pair of users is higher than 0.3, which is the threshold used to recognize users with close social relationships. We call a group of users where each pair of users have a close relationship a "clique". Finding all cliques in a graph is also a well-known NP-complete problem.

We take an iterative procedure to get all cliques. Specifically, we first generate a corresponding graph according to the procedure described before. Then, iteratively, we pick a maximum clique each time in the graph and delete all vertices in the clique and all corresponding edges from the graph until there are no more vertices left in the graph. The reason that we pick the maximum clique each time is that the order in which we remove cliques from the graph does not change an original clique from being a clique in the left graph, removing the max clique each time helps shrink the size of the graph, which decreases the complexity of the S^3 problem. If there are multiple maximum cliques found, we choose the one with the largest sum of edges. The reason is that a larger sum of edges means those users are more likely to leave and they need to be distributed to different APs. After find a user clique, we distributed them and find the next clique until no users need to be allocated.

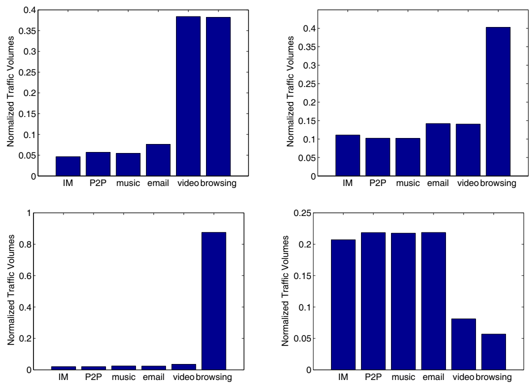

Fig. 8. Cluster centroid of four user groups

To pick a maximum clique from the graph, we adopt a heuristic branch-and-bound algorithm [25]. Each time the users are first sorted by a greedy vertex coloring algorithm. Then, the search starts from the first vertex.

B. AP Selection Algorithm

In this subsection, we describe the AP selection algorithm adopted in S^3. We start from the design principles and then present the detailed algorithm.

1) Design principles: In general, the problem is to distribute users to a set of different APs so that the sum of the social relation index δ between each pair of users on each AP under a controller is minimized. Towards this goal, user pairs with tight δ should be dispersed to different APs. For this purpose, we augment users to APs so that the increment on the total δ is minimal.

2) AP selection algorithm: For ease of presentation, we use the following notations. Let $S(AP_i), i = 1, ..., n$ be the set of users associated in the ith AP. Fig. 9 shows the pseudopod of the AP selection algorithm. The AP selection algorithm will output the ID of an AP, say $1 \leq j \leq n$, to a new user u. Specifically, if $S(AP_i)$ is empty or there are multiple candidate APs to choose, we simply apply LLF. Let $T(AP_i) = \Sigma_{u \in S(AP_i)} w(u)$ be the traffic at the AP_i and $C(AP_i)$ be the total social relation index when u is added to AP_i, i.e.,

$$C(AP_i) = \sum_{\forall w \in S(AP_i)} \delta(u, w).$$

Notice that the cost will be set as infinite if the bandwidth constraint cannot be satisfied (line 8 and 9 in Fig. 9). The demand of each user bandwidth $W(u)$ can be estimated using the history trace of u as studied in work [10].

C. Discussion

The AP selection algorithm tries to associate new user to the AP so that the increment of the total internal social relations within the domain of a controller is minimized. The rational behind this strategy is to assign users with strong social relationships to different APs since they tend to leave together, which may cause unbalanced workload of APs. It is clear that when users have strong social relationships, our algorithm will achieve great gain in balancing AP workload. It is often the case for most enterprise WLANs since colleagues have regular working schedules, e.g., meetings. In addition, our algorithm fits enterprise WLANs most since it can be implemented on WLAN controllers, where most recent historical login records of users can be collected and used for extracting social relation indexes between users. We will further investigate how parameters can be chosen for our online algorithm in the performance evaluation section via extensive trace-driven simulations.

Algorithm 1 Social Relation Based Selection

Input:
$\delta(u, v)$ for all users u and v
User u to be allocated
Output:
$S(AP_i)$
1: Initial $T(AP_i)$: $\sum_{i=1}^{n} \Sigma_{v \in AP_i} w(v)$
2: Initial $G(V, E)$(V={users to be distributed}, E={social relation between users over 0.3})
3: **while** G$\neq \emptyset$ **do**
4: Find clique G_c from G
5: Search the solution space of distribution users in G_c and sort them by $\sum_{i=1}^{n} C(AP_i)$
6: Find the top 30% distribution and choose the one which make β min
7: Update $S(AP_i)$
8: erase G_c from G
9: **end while**
10: **return** $S(AP_i)$

Fig. 9. The pseudocode of AP selection algorithm

V. Performance Evaluation

A. Methodology

We evaluate our S^3 AP-selection algorithm based on trace-driven simulations. We use the same trace as described in Section II and use four-week trace data from July 4 to July 24, 2012 as the learning stage for establishing social relationships between users, leaving the trace data from July 25 to July 27 for AP selection experiments. We compare S^3 AP-selection algorithm with LLF , where a new coming user is allocated to the AP with the least workload. We consider the balance index of throughput among all APs in WLAN controller domains to evaluate the performance of our S^3 and the LLF algorithms.

B. Effect of Algorithm Parameters

In this experiment, we examine the effect parameters involved in the algorithm. We first look at the time window size used to extract co-leaving events. We change the length of the co-leaving extraction time interval from one minute to twenty minutes with a step of five minute to extract co-leaving

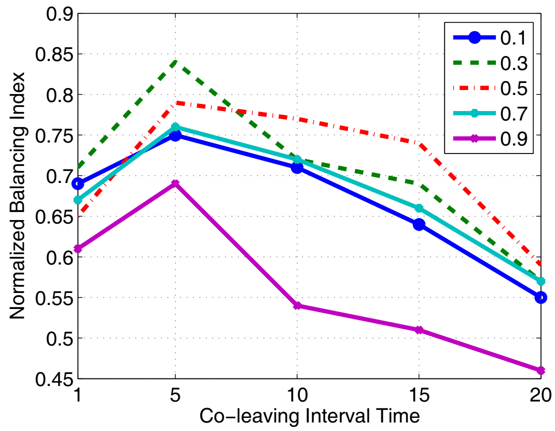

Fig. 10. The balancing index varying under two parameters in simulation

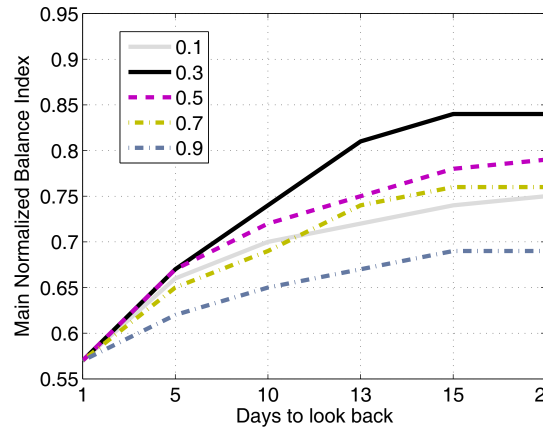

Fig. 11. The balancing index varying under two parameters in simulation

events in the trace data. Fig. 10 plots the average normalized balancing index over all WLAN controllers as a function of co-leaving extraction time intervals. It can be seen that, as the length of the extraction time interval increases, the normalized balancing index first increases, reaches to a maximum at the optimal time interval length of five minutes, and then drops. This trend can also be seen using different constant coefficient α in calculating the balancing index. The reason is that a larger time interval will generate more co-leaving events, which facilitates the calculation of social relationship strength between users. On the other hand, it will also introduce more fake social relationships as the chance for two non-related users to share a common co-leaving event increases. In contrast, a smaller time interval can result more accurate social relationships but the number of such relationships would be limited, which is important to S^3 algorithm. We then use $\alpha = 0.3$ and a five-minute co-leaving extraction time interval as the optimal parameter configuration.

We then examine whether the parameter τ can represent the real social relation and achieve the best result. We used different training days and test days to estimate this parameter.We found that the value of system parameters we choose before always lead to a higher normalized balance index among APs.

We then investigate how much history data do we need. In this experiment, we use five minutes co-leaving extraction time and varies the value of α. Fig. 11 plots the average normalized balancing index over all WLAN controllers as a function of historical data. It can be seen that, for all α, as more historical data are available, the normalized balancing index increases and stabilizes when the length of training stage reaches about 15 days. This implies that information older than 15 days does not help in increasing the balance index but does not hurt either.

C. Comparison with LLF

In this section, we compare S^3 with LLF .We take all training data for establishing pairwise social relationships and use the same experimental data for all the algorithm to assign users to APs. Fig. 12 shows the average normalized balancing index over all WLAN controllers and all experimental data as a function of time in daytime. There are two main observations found in Fig. 12. First, it can be seen that S^3 outperform LLF over most time. On average, S^3 can achieve about 41.2% balancing index gain compared with LLF. Second, the performance of S^3 is more stable and robust against user behavior than that of LLF. Especially, S^3 performs well when suffering co-leaving events. For example, in SJTU, time from 12:00 to 13:00 , from 16:00 to 17:50 and 21:00 to 22:00 are peak time when users leave network, against which S^3 can achieve about 52.1% balancing index gain against LLF. The reason is that s^3 can effectively cancel the negative impacts of social relations on AP load balance.

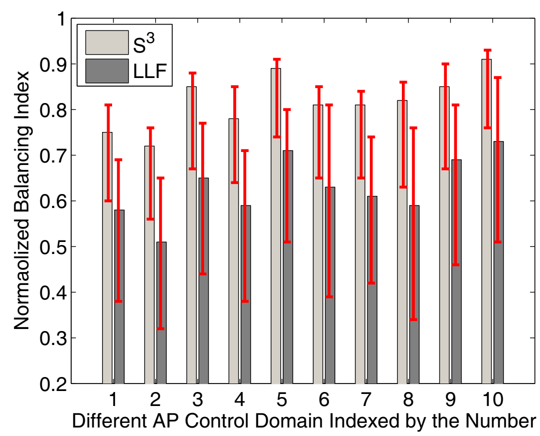

Fig. 12. Comparison between S^3 and LLF in one controller domain

To further demonstrate the effectiveness of S^3 on all sites, Fig. 12 shows the average normalized balancing index on all sites with 95% percentage confidence error bar. We find that S^3 shows much higher stability than LLF. The error bar can be reduced by 72.1% overall compared with LLF. The average normalized balancing index can be improved by 41.2% overall.

These results demonstrate that S^3 effectively distribute users to APs, providing significant improvements in balancing AP loads.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have studied the load balancing problem in enterprise WLANs. By systematically mining the WLAN trace we collected,we have found the fundamental leading factors to load balancing problem is the churns of WLAN users. We have also observed obvious social behavior between users which can be characterized with application profiles of users. With this insight, we have proposed an innovative scheme S^3 for user-friendly and steady load balancing in enterprise WLANs. S^3 is resilient to churns of WLAN users and maintain excellent user experience without requiring to migrate user among APs. The real prototype implementation have verified the feasibility of the S^3 design. Moreover, extensive accurate trace-driven simulations have also demonstrated the efficacy of S^3. In our future work, we will further examine more aspects in characterizing the network usage profiles of users so that they can be used to obtain more accurate sociality information of users. In addition, we will implement S^3 in our campus WLAN and further improve the S^3 design by solving the issues encountered in practice.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant No.61170237 and No.61202375, the National High Technology Research and Development Program under Grant No.2011AA010502 and STCSM Grant 12ZR1414900.

REFERENCES

[1] J. Erman, A. Gerber, M. T. Hajiaghayi, D. Pei, and O. Spatscheck. Network-aware forward caching. In WWW, 2009.

[2] Q. Xu, A. Gerber, Z. M. Mao, J. Pang, and S. Venkataraman. Identifying diverse usage behaviors of smartphone apps. In ACM IMC, 2011.

[3] Spearman C. "The proof and measurement of association between two things". Amer. J. Psychol. 15(1904): 72-101.

[4] J. MacQueen. Some methods for classification and analysis of multivariate observations. In Fifth Berkeley Symposium on Math Statistics and Probability, 1967.

[5] M. M.-T. Chiang and B. Mirkin. Experiments for the number of clusters in k-means. In Lecture Notes in Computer Science, Progress in Artificial Intelligence, 2007.

[6] J. Hartigan. Clustering Algorithms. J. Wiley & Sons, 1975.

[7] B. Mirkin. Clustering for Data Mining: A Data Recovery Approach. Chapman and Hall, 2005.

[8] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63:411C423, 2001.

[9] G. Judd and P. Steenkiste, Fixing 802. 11 access point selection, ACM SIGCOMM Computer Communication Review, vol. 32, no. 3, pp. 31C31, 2002.

[10] Y. Qiao, J. Skicewicz, and P. Dinda, An empirical study of the multiscale predictability of network traffic, in High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on. IEEE, 2004, pp. 66-76.

[11] J. Kleinberg, E. Tardos, and Y. Rabani, Fairness in routing and load balancing, in focs. Published by the IEEE Computer Society, 1999, p. 568.

[12] D. Rathunde, Dynamic load balancing during message processing in a wireless communication service network, Jun. 3 2003, uS Patent 6, 574, 477.

[13] X. Wan, X. Wang, U. Heo, and J. Choi, A new ap-selection strategy for high density ieee802. 11 wlans, in 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE, 2010, pp. 52-58.

[14] A. Nicholson, Y. Chawathe, M. Chen, B. Noble, and D. Wetherall, Improved access point selection, in Proceedings of the 4th international conference on Mobile systems, applications and services. ACM, 2006, pp. 233-245.

[15] I. Jabri, N. Krommenacker, T. Divoux, and A. Soudani, Ieee 802.11 load balancing: an approach for qos enhancement, International Journal of Wireless Information Networks, vol. 15, no. 1, pp. 16-30, 2008.

[16] Y. Bejerano and S. Han, Cell breathing techniques for load balancing in wireless lans, IEEE Transactions on Mobile Computing, pp. 735-749, 2009.

[17] H. Velayos, V. Aleo, and G. Karlsson, Load balancing in overlapping wireless lan cells, in Communications, 2004 IEEE International Conference on, vol. 7. IEEE, 2004, pp. 3833-3836.

[18] R. Akl and S. Park, Optimal access point selection and traffic allocation in ieee 802.11 networks, Proceedings of 9th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2005): Communication and Network Systems, Technologies and Applications, vol. 8, pp. 75-79, 2005.

[19] F. Xu, C. Tan, Q. Li, G. Yan, and J. Wu, Designing a practical access point association protocol, in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1-9.

[20] E. Garcia, R. Vidal, and J. Paradells, Cooperative load balancing in ieee 802.11 networks with cell breathing, in Computers and Communications, 2008. ISCC 2008. IEEE Symposium on. IEEE, 2008, pp. 1133-1140.

[21] X. Chen, Y. Zhao, B. Peck, and D. Qiao, Sap: Smart access point with seamless load balancing multiple interfaces, in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1458-1466.

[22] K. Wu, H. Li, L. Wang, Y. Yi, Y. Liu, Q. Zhang, and L. M. Ni, hjam: Attachment transmission in wlans, in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1449-1457.

[23] W. Hsu, D. Dutta, and A. Helmy, Mining behavioral groups in large wireless lans, in Proceedings of the 13th annual ACM international conference on Mobile computing and networking. ACM, 2007, pp. 338-341.

[24] W. Hsu and A. Helmy, On nodal encounter patterns in wireless lan traces, in Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on. IEEE, 2006, pp. 1-10.

[25] P.R. O stergard, A Fast Algorithm for the Maximum Clique Problem, Discrete Applied Math., vol. 120, nos. 1-3, pp. 197-207, 2002.

[26] Chiu, Dah-Ming; Raj Jain (1989). "Analysis of increase and decrease algorithms for congestion avoidance in computer networks". Computer Networks and ISDN systems 17: 1C14.