Characterizing Sociality for User-Friendly Steady Load Balancing in Enterprise WLANs

Guangtao Xue, Yanmin Zhu, Zhenxian Hu, Hongzi Zhu, Chaoqun Yue and Jiadi Yu

Abstract

Traffic load is often unevenly distributed among the access points in enterprise WLANs. Such load imbalance results in sub-optimal network throughput, unfair bandwidth allocation among users, and unsatisfactory user quality of experience. We have collected real traces from over 12,000 WiFi users at Shanghai Jiao Tong University. Through intensive data analysis, we find that the social behavior of users (e.g., leaving together) may cause a significant AP load imbalance problem. We also observe from the traces that users with similar application usage have the potential to leave together. Inspired by those observations, we propose a 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 to different APs based on the obtained knowledge. Trace-driven simulation results show that S^3 is feasible and can achieve better balancing performance when compared to state-of-the-art balance algorithms.

In recent years, IEEE 802.11 wireless LANs (WLANs) have been widely deployed in businesses, public areas, and homes. Studies [1–3] on operational WLANs have shown that the traffic load is often unevenly distributed among 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 in providing maximum throughput and fair services to its users. It dramatically degrades WiFi user quality of experience (QoE) [4, 5].

To solve the load balancing problem and satisfy WiFi user QoE, 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 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 [4], 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 are incapable of 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, these schemes can achieve good load balancing performance but also cause unpleasant constant connection disruptions. As a result, there is no existing scheme, to the best of our knowledge, that can successfully tackle the load balancing problem in enterprise WLANs and achieve superior load balancing while still preserving good user experience. Recent work [6] has analyzed the high predictability of human behavior, which is mostly driven by regular routine activities in mobile networks.

In this article, we use an empirical methodology to study the load balancing problem in enterprise WLANs. We have collected real WLAN traces from more than 12,000 users at Shanghai Jiao Tong University 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 percent peak time and 60 percent off-peak time, traffic load on APs is rather unbalanced in our trace using the least loaded first (LLF) scheme [7].

We further determine the principal 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, that is, 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 usage profiles.

With this insight, we propose a sociality-aware AP selection scheme, 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 usage profiles. With the knowledge of application usage profiles and social relationships of users, S^3 elegantly distributes co-leaving users

The authors are with Shanghai Jiao Tong University.

 0890-8044/15/$25.00

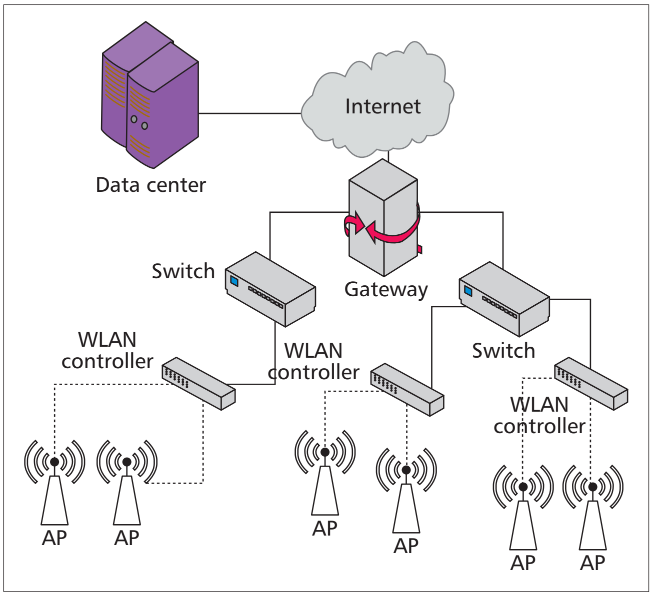

Figure 1. Structure of the WLAN in SJTU.

to a set of APs considering the current workload on those APs as well. The main advantage of S^3 is two-fold. First, S^3 is user-friendly as it does not migrate users from one AP to another. Second, it is very resilient to network churns as it can resist sudden traffic demand changes caused by co-leaving. Through extensive accurate trace-driven simulations, we demonstrate the efficacy of S^3.

Empirical Analysis Based on Real WLAN Traces

Collecting WLAN Usage Traces

We collect WLAN usage trace data from Shanghai Jiao Tong University (SJTU), a prestigious university in the mainland of China. Figure 1 illustrates a typical enterprise WLAN deployed at SJTU, which consists of three major entities: lightweight APs, WLAN controllers, and a back-end data center. A WLAN controller taking charge of several APs in the vicinity is responsible for assigning users to specific APs within its domain. The state-of-the-art strategy adopted by a controller is to assign a new user to the AP with the least traffic load (or the lowest number of users). These controllers are default infrastructure in WLANs. There are 334 APs in the campus-wide WLAN. All the APs are physically connected to the campus core network, and each AP is logically connected to one of the controllers via tunneling. We collect traces from the back-end data center, which records all login information.

For the study in this article, we collected trace data over three months from July to October in 2012, which involved 12,374 users collected from 334 APs deployed in 22 buildings. The specific fields in logged records include user identifiers (i.e., medium access control, MAC, addresses of wireless cards), connected timestamps (the time instances when users successfully connected to an AP), disconnected timestamps (the time instance when users disconnected from an AP), accessed APs, and the served traffic amounts (the total traffic amount users sent to or received from an AP during a connection). In addition, from the core network routers, we also obtained all WLAN traffic information, including the source and destination IP addresses of a packet, and transportation layer and application layer ports (e.g., tcp, HTTP, DNS, SIP). By analyzing the port combination using certain heuristics [8], concrete applications can be accurately identified. In our traces, all user identifiers were processed with hash functions (e.g., SHA) to remove privacy. For proprietary reasons, the results presented in this article 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 traces, 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. Furthermore, we categorize these top applications into the following six application realms: instant messaging, peer-to-peer, music, email, video, web browsing.

Load Imbalance 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 traces. To better quantify the load balance level among a set of APs, we use the following balancing index definition [9]:

Definition: Given n APs, let T_i denote the throughput of the i-th AP, $i = 1, \ldots, n$, 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

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

This index has been widely used in the literature to assess load balancing performance. Other fairness metrics, such as *max-min* [10] and *proportional fairness* [11], may also be used. With different load-balancing strategies, the balancing index ranges from $1/n$ to 1 with larger index value indicating better balancing level. Figure 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 percent of the time during peak hours and about 60 percent of the 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-the art AP selection strategy (i.e., LLF) [7].

Principal 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.

Factor of Application Dynamics: To know whether the application accounts change, 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 index among all APs under a controller in the ith sub-time-period, denoted

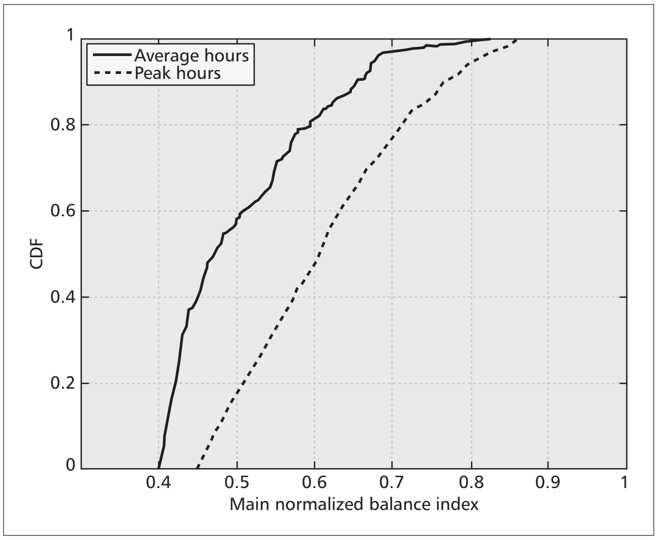

Figure 2. CDF of the normalized balance index over all controllers

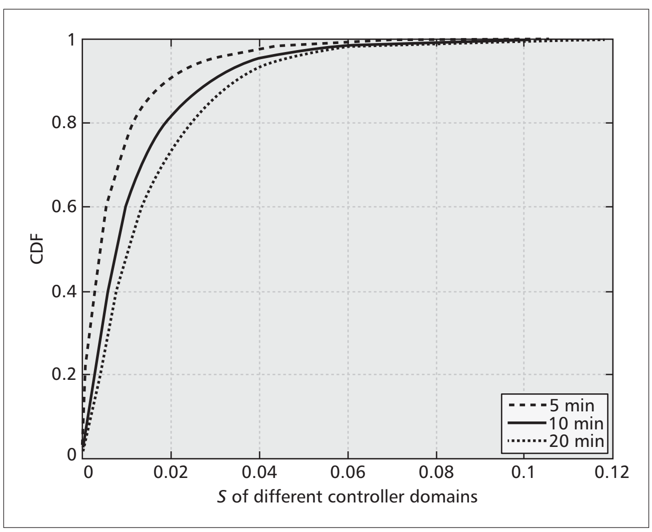

Figure 3. CDF of the variance of balance index over all controllers.

as β_i, and calculate the variance of β_i, i = 1, …, n. Figure 3 plots the CDFs of the variance of balance index over all time periods and all controllers with the length of a sub-time-period equal to 5 min, 10 min, and 20 min, respectively. It can be seen that more than 80 percent of variance is less than 0.02 with 10-min sub-time-periods. This result shows that the balance index does not change suddenly with fixed users.

Factor of User Dynamics: We then investigate the significance of user dynamics in the load imbalance problem. We first check the relationship between the number of users and the throughput of an AP. Figure 4 shows an example of an AP where the number of users and the throughput of the AP are shown on a workday from 8:00 to 24:00. It can be seen that there is strong correlation between the two. To better quantify the statistical dependence between the number of users on an AP and the throughput of the AP, we use Spearman Rank [12], which can assess how well the relationship between two variables can be described using a monotonic function. The Spearman's rank correlation coefficient of Fig. 4 is 0.7263, which indicates that there is strong correlation between the number of users and throughput of that AP.

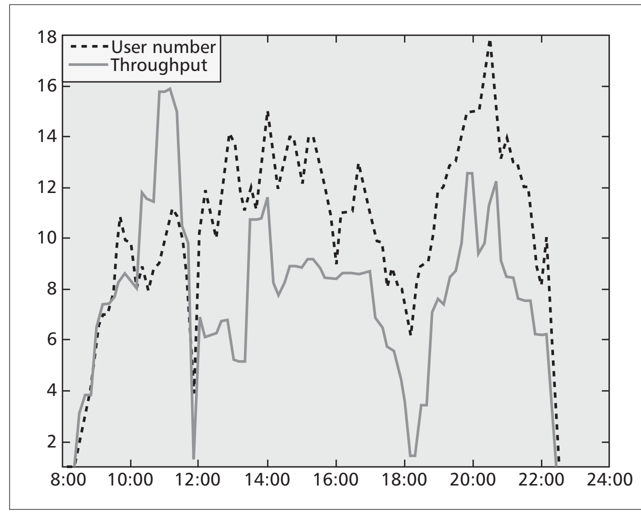

Figure 4. An example of the relationship between the number of users and the traffic throughput of an AP.

Enlightened by this example, we further examine whether all APs show similar results. For each AP, we calculate the Spearman's rank correlation coefficient for each day, using a time window of 10 min, 20 min, and 30 min, respectively, to check the number of users and the traffic amount. Figure 5 plots the CDF of the correlation coefficient over all APs. It can be seen that almost 80 percent of the coefficient is higher than 0.8, which implies that the throughput of an AP is highly correlated with the number of users associated with that AP.

We now investigate the impact of churns of users on the load balancing problem of WLANs. We use a similar method to quantify the balance index of the number of users β_{num} among all APs under a WLAN controller. Figure 6 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 the 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 LLF, where a newly arriving user is allocated to the AP with the lowest workload, 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 imbalance. In this case, LLF cannot recover from such load imbalance. Therefore, we conclude that the churn of users, especially co-leaving of users, plays an essential role in causing load imbalance in enterprise WLANs.

Revealing the User Sociality behind Churns

Sociality of WLAN Users

In this section, we investigate user activities that might cause unbalanced leaving. For example, people in an enterprise domain often have routine activities, such as classes at schools and department meetings in corporations. These social activities may have great influence on the way people use a WLAN. From the perspective of AP accessing behavior, we study two main events in the trace data that may reflect those social activities:

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

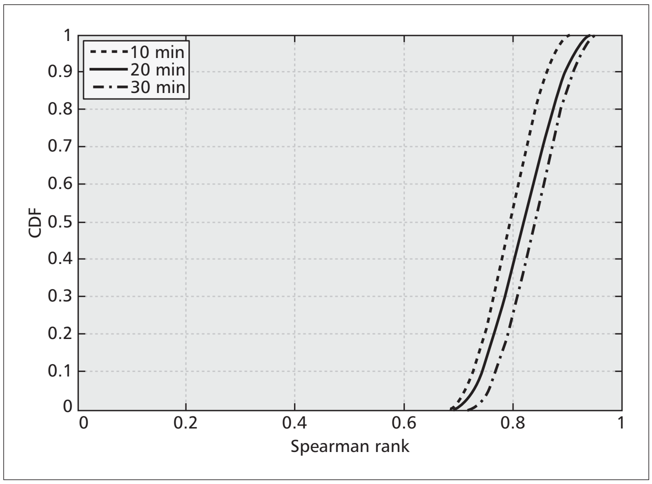

Figure 5. CDF of correlation coefficients over all APs.

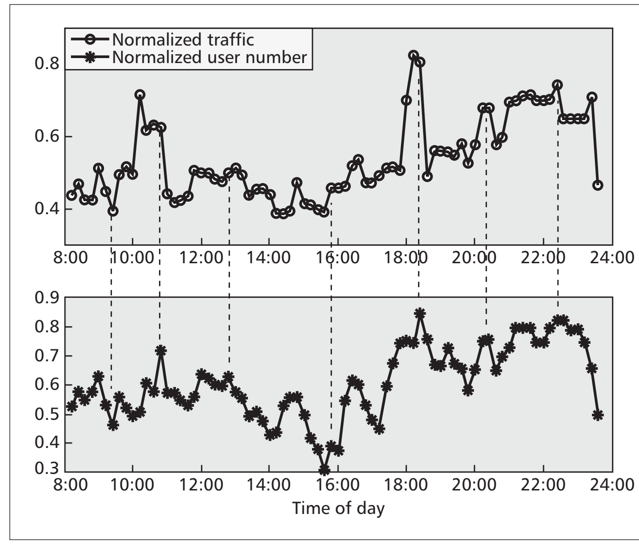

Figure 6. An example of the relationship between the balance index of the number of users and the balance index of traffic load.

Two users are said to have a social relationship if they share the aforementioned events in common. 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 period of time is used to extract co-leaving events. Such fake social relationships are random and have no ability 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. 7, using three different periods of time for event extraction: 1 min, 10 min, and half an hour. It indicates that most users show strong sociality in their AP access behavior and do not leave an AP independently.

Capturing User Sociality

We find that users who have higher probability of leaving together also have more similar application profiles. Inspired by this connection, we further investigate whether two users who share similar application usage profiles 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. The k-means algorithm is a simple but effective technique to cluster feature vectors into a predefined k number of groups [13]. The selection of an appropriate value of k is crucial and is an open research issue. We use one of the most well-known heuristics, called gap statistic [14], to find the optimal value of k. After selecting the optimal 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 regarding the optimal clustering result, we plot the cluster centroid of four user groups, as Fig. 8 shows. We observe that a user can be divided into a distinct group according to its application usage profile. We label these four groups *type1*, *type2*, *type3*, and *type4*. Let τ ($type_i$, $type_j$) 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 by τ ($type_i$, $type_j$) having greater values in the diagonal line of the table). We consider utilizing this strong pattern for forecasting co-leaving events.

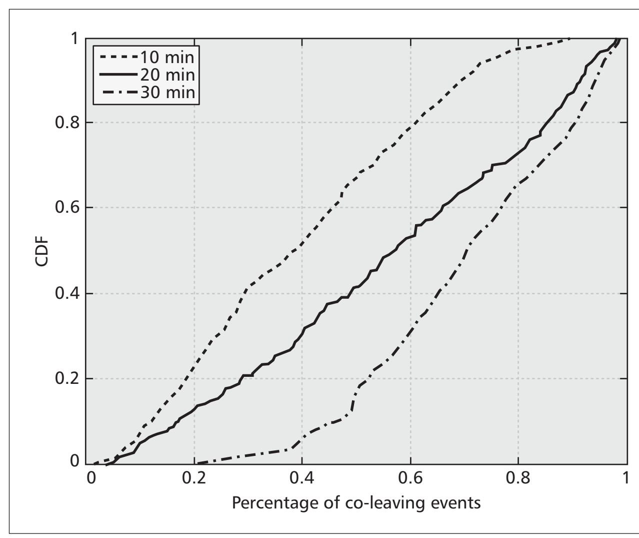

Figure 7. CDF of the number of co-leaving events to the total number of leaving events over all users.

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. Even without some mobile users' sociality behaviors knowledge, we can use user application usage profiles to predict social behavior of users such as co-leaving. With this intuition, we can distribute user pairs of tighter social relations to different APs so that users in the same APs are diversified. It is very resilient to network churns as it can resist sudden traffic demand changes caused by co-leaving.

Social-Aware AP Selection

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. Lastly, we present our social-aware AP selection algorithm for enterprise WLANs.

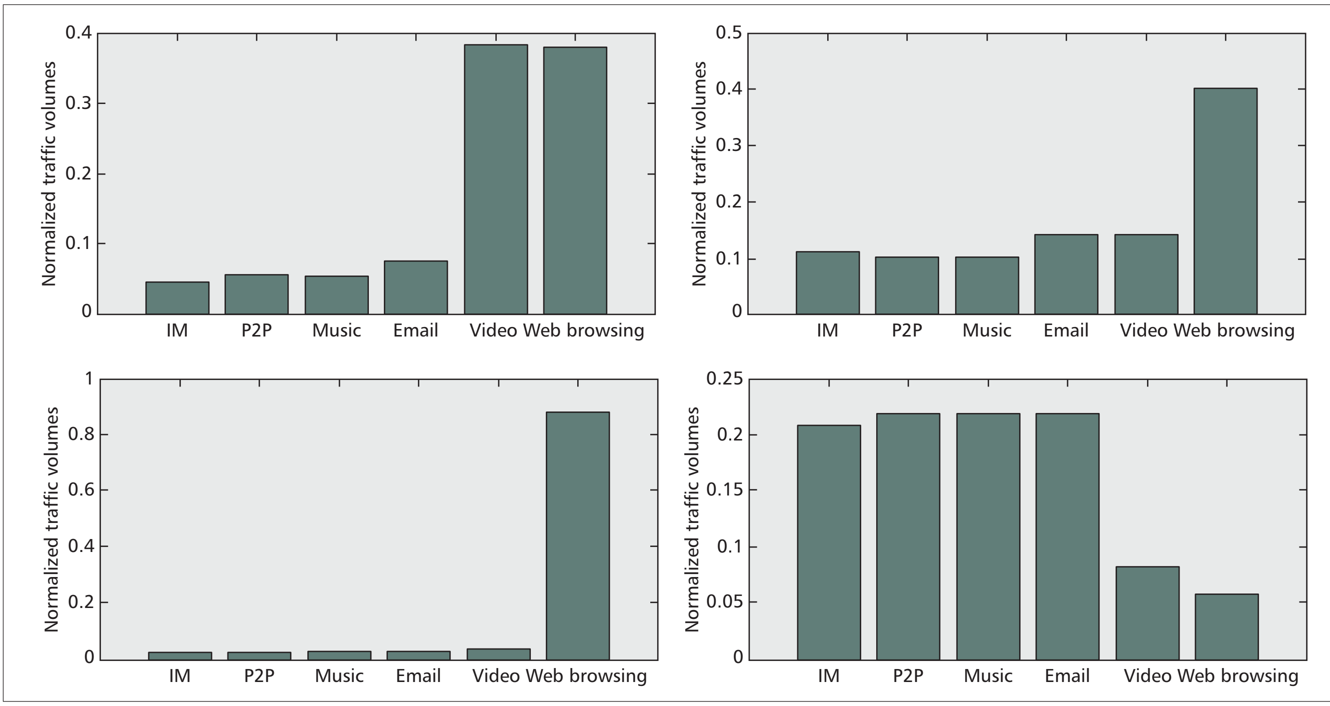

Figure 8. Cluster centroid of four user groups.

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

Table 1. Possibility of leaving together between different usage types.

Given two users u and v, their social index 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 each other 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 a constant coefficient. Thus, a high social relation index implies a stronger relation between users and vice versa. We expect the social relation index to effectively forecast the co-leaving events between users that affect the balanced index in the same controller domain.

Problem Statement

The original AP balancing problem is to distribute the subscribing users to different APs so that all the APs are kept balanced at all times, that is, $\min(\Sigma\beta)$. In practice, however, this problem has no optimal solution because the optimal solution requires the exact leaving time of each user. Nevertheless, such leaving time information is of the future and can never be obtained. Fortunately, we recall that the main cause of the AP imbalance is the user co-leaving events.Therefore, we take another approach toward deriving the optimal AP balance solution. 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 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), $u, v \in \Sigma_{i=1}^{n} AP_i$. 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:

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

subject:

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

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

The problem proposed here is also a multi-objective optimization problem. To achieve the solution, we consider this object as the main object to achieve. As our scheme makes the balance index not too bad due to the erasure 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 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 complicity 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 finding a user clique, we distribute them and find the next clique until no users need to be allocated.

To pick a maximum clique from the graph, we adopt a heuristic branch-and-bound algorithm [15]. Each time the users are first sorted by a greedy vertex coloring algorithm. Then the search starts from the first vertex.

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. Toward 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 $S(AP_i)$, $i = 1, \ldots, n$ be the set of users associated in the ith AP. The AP selection algorithm will output the ID of an AP, say $1 \le j \le n$, 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, that is,

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

Notice that the cost will be set as infinite if the bandwidth constraint cannot be satisfied. The demand of each user bandwidth $W(u)$ can be estimated using the history trace of u as studied in [16].

Prototype Implementation

We have implemented our scheme on the first floor of the SEIEE Building at SJTU. More specifically, we deploy four APs and a server emulating the controller. Figure 9 shows the layout of the settings. The server actively monitors all traffic of connected users and runs the S^3 algorithm. To assign users to different APs, we have developed a client software to connect with the server. Users have to use the client to access the AP. When the client starts, it first accesses one of the APs randomly to connect to the server. Then it sends a user's identity to the server, which is calculated by the user's MAC address. When the server receives the message sent by the client, it first clusters the user to one of the usage pattern groups. Then it calculates the social index with the accessed user. After that the server runs the distribution algorithm and returns the result to the client. Then the client can access the appropriate AP.

We conducted a small-scale real experiment as follows. In the first two weeks, we applied LLF on the server for the purpose of comparison. The server uses Simple Network Management Protocol (SNMP) to monitor the load of APs, which makes it possible to run LLF without a real WLAN controller. During these two weeks, network traffic was collected for training our algorithm using LLF to assign users. In the third week, the server started to run S^3. The social index between each pair of users was calculated. We then analyzed the results of the experiment. Figure 10 plots the normalized balancing index with 95 percent confidence bar of both LLF and S^3 during the three-week experiment. It can be seen that S^3 can achieve better load balancing. On average, S^3 can achieve about 36.8 percent load balancing gain compared to LLF. Furthermore, S^3 is also more stable than LLF, which indicates that S^3 can evenly distribute users with close social relationships among the four APs.

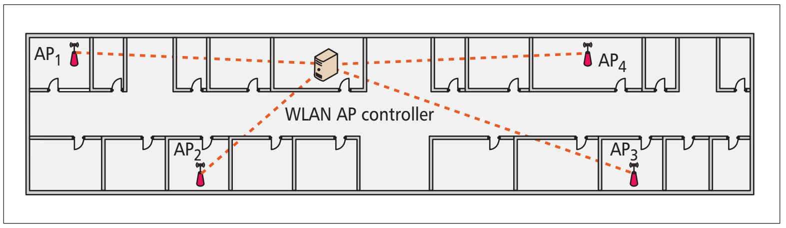

Figure 9. Layout of the prototype testbed.

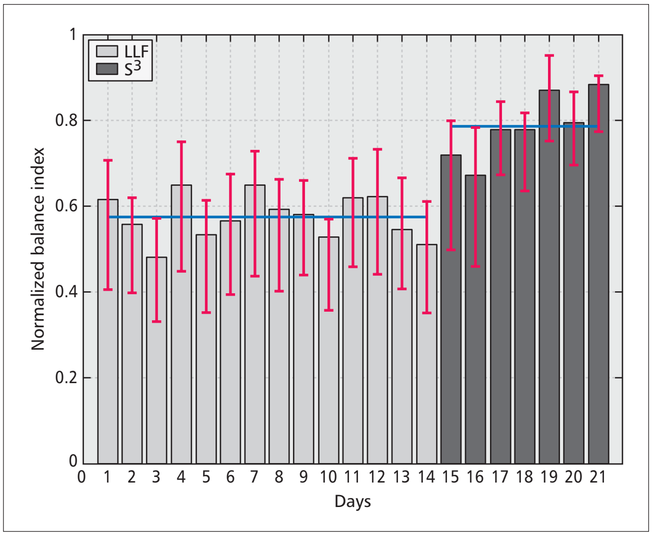

Figure 10. Result of the prototype.

Although we use a server to emulate a real WLAN controller, the experiment results show that S^3 is feasible in practice. From the experience of the prototype implementation, we have learned that S^3 can achieve rather steady load balance while keeping all established links up. We further evaluate the performance of S^3 through extensive trace-driven simulations in the next section.

Performance Evaluation

Methodology

We evaluate our S^3 AP-selection algorithm based on trace-driven simulations. We use the same traces described earlier 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 the S^3 AP-selection algorithm with LLF and another prior scheme called social relation distribution (SRD) [17], which only uses co-leaving events to characterize the sociality of users without considering user application usage profiles. We consider the balance index of throughput among all

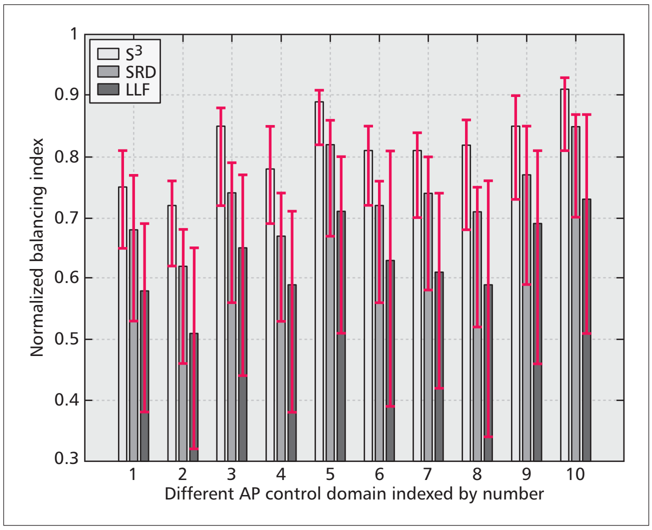

Figure 11. Comparison between S^3 and LLF and SRD.

APs in WLAN controller domains to evaluate the performance of our S^3 and the LLF and SRD algorithms.

Comparison with LLF and SRD

Here, we compare S^3 with LLF and SRD.We take all training data for establishing pairwise social relationships and use the same experimental data for all the algorithms to assign users to APs. Figure 11 shows the average normalized balancing index with a 95 percent confidence error bar over all WLAN controllers and all experimental data as a function of time in daytime. There are two main observations found in Fig. 11. First, it can be seen that S^3 outperforms both LLF and SRD over most time. On average, S^3 can achieve about 41.2 and 22.4 percent balancing index gain compared to LLF and SRD, respectively. Second, the performance of S^3 is more stable and robust against user behavior than that of LLF. In particular, S^3 performs well when suffering co-leaving events. For example, at SJTU, from 12:00 to 13:00, from 16:00 to 17:50, and from 21:00 to 22:00 are peak times when users leave the network, for which S^3 can achieve about 52.1 percent balancing index gain against LLF. The reason is that S^3 can effectively cancel the negative impacts of social relations on AP load balance. These results demonstrate that S^3 effectively distributes users to APs, providing significant improvements in balancing AP loads and WiFi user QoE.

Conclusion

In this article, we have studied the load balancing problem in enterprise WLANs. By systematically mining the WLAN traces we collected, we have found the fundamental principal factors of the load balancing problem are the churns of WLAN users. We have also observed obvious social behavior between users that 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 maintains excellent user experience without requiring migrating the user among APs. The real prototype implementation has verified the feasibility of the S^3 design. Moreover, extensive accurate trace-driven simulations have also demonstrated the efficacy of S^3.

Acknowledgment

This work was supported by the National High Technology Research and Development Program of China (2015AA01A202), NSFC Grants (61572324, 61170237).

References

[1] X. Wan *et al.*, "A New AP-selection Strategy for High Density IEEE 802.11 WLANs," *2010 Int'l. Conf. Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2010, pp. 52–58.
[2] K. Wu *et al.*, "hJam: Attachment Transmission in WLANs," *IEEE Trans. Mobile Computing*, vol. 12, no. 12, 2013, pp. 2334–45.
[3] K. Wu *et al.*, "Side Channel: Bits over Interference," *IEEE Trans. Mobile Computing*, vol. 11, no. 8, 2012, pp. 1317–30.
[4] I. Jabri *et al.*, "IEEE 802.11 Load Balancing: An Approach for QoS Enhancement," *Int'l. J. Wireless Info. Networks*, vol. 15, no. 1, 2008, pp. 16–30.
[5] M. Dong *et al.*, Quality-of-Experience (QoE) in Emerging Mobile Social Networks," *IEICE Trans. Info. and Sys.*, vol. 97, no. 10, 2014, pp. 2606–12.
[6] P. O. S. V. De Melo *et al.*, "Recast: Telling Apart Social and Random Relationships in Dynamic Networks," *Performance Evaluation*, vol. 87, 2015, pp. 19–36.
[7] G. Judd and P. Steenkiste, "Fixing 802. 11 Access Point Selection," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 32, no. 3, 2002, p. 31.
[8] J. Erman *et al.*, "Network-Aware Forward Caching," *WWW*, 2009.
[9] D. M. Chiu and R. Jain, "Analysis of Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, 1989, pp. 1–14.
[10] Y. Bejerano and S. Han, "Cell Breathing Techniques for Load Balancing in Wireless LANs," *IEEE Trans. Mobile Computing*, vol. 06, no. 09, 2009, pp. 735–49.
[11] J. Kleinberg, E. Tardos, and Y. Rabani, "Fairness in Routing and Load Balancing," *J. Comp. and Sys. Sciences*, vol. 63, no. 1, 2001, pp. 2–20.
[12] C. Spearman, "The Proof and Measurement of Association between Two Things," *American J. Psychology*, vol. 15, no. 1, 1904, pp. 72–101.
[13] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *5th Berkeley Symp. Math Statistics and Probability*, 1967, pp. 281–97.
[14] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Data Set via the Gap Statistic," *J. Royal Statistical Soc.: Series B (Statistical Methodology)*, vol. 63, no. 2, 2001, pp. 411–23.
[15] P. R. O Stergard, "A Fast Algorithm for the Maximum Clique Problem," *Discrete Applied Mathematics*, vol. 120, no. 1–3, 2002, pp. 197–207.
[16] Y. Qiao, J. Skicewicz, and P. Dinda, "An Empirical Study of the Multiscale Predictability of Network Traffic," *Proc. 13th IEEE Int'l. Symp. High Performance Distributed Computing*, 2004., pp. 66–76
[17] G. Xue *et al.*, "Sociality-Aware Access Point Selection in Enterprise Wireless LANs," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 10, 2013, pp. 2069–78.

Biographies

GUANGTAO XUE [M] obtained his Ph.D. from the Department of Computer Science and Engineering at Shanghai Jiao Tong University, China, in 2004. He is an associate professor in the same department. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and distributed computing. He is a member of the IEEE Computer and Communications Societies.

YANMIN ZHU [M] obtained his Ph.D. from the Department of Computer Science and Engineering at Hong Kong University of Science and Technology in 2007. He is an associate professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. Prior to joining Shanghai Jiao Tong University, he worked as a research associate with the Department of Computing at Imperial College London, United Kingdom. His research interests include vehicular networks, wireless sensor networks, and mobile computing. He is a member of the IEEE Communications Society.

ZHENXIAN HU [S] is currently a Ph.D. candidate in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. He received his M.S. degree in communication systems from China Academy of Telecommunication Technology in 2011. His research interests include computer network protocols and systems, wireless networks, mobile computing, and network measurement.

HONGZI ZHU [M] obtained his Ph.D. from the Department of Computer Science and Engineering at Shanghai Jiao Tong University in 2009. He is an associate professor in the same department. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and network security. He is a member of the IEEE Computer and Communications Societies.

CHAOQUN YUE [S] received his M.S. degree from the Department of Computer Science and Engineering at Shanghai Jiao Tong University in 2014. His research interests include wireless network and mobile computing.

JIADI YU is an assistant professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He obtained his Ph.D. from the same department in 2007. In the past, he worked as a postdoctoral researcher at Stevens Institute of Technology, New Jersey, from 2009 to 2011. His research interests include networking, mobile computing, cloud computing, and wireless sensor networks.