Rethinking Mobile Devices' Energy Efficiency in WLAN Management Services

Haoxin Wang and Jiang Xie
University of North Carolina at Charlotte
Email: {hwang50, Linda.Xie}@uncc.edu

Xingya Liu
Lamar University
Email: xliu@lamar.edu

***Abstract*—With the rapid popularization of large data stream mobile applications, wireless local area networks (WLANs) have been a top choice for mobile users (MUs), because of the high data rate and low monetary cost. However, the battery life of mobile devices, which is the most concerned feature of MUs, may suffer from WLAN management services, such as mobility management and load balancing services. Unfortunately, few existing WLAN systems take into account both the energy efficiency of mobile devices and the performance of management services. Even worse, to improve the performance of WLAN management services, various existing management mechanisms sacrifice mobile devices' energy. In this paper, we propose BELL, a novel WLAN system that provides two energy-efficient management services for its associated MUs by reproducing and scheduling the beacons broadcast from access points (APs). We name them BELL-handoff and BELL-2M services. We have implemented the proposed BELL-handoff using commercial Wi-Fi adapters. The experimental results reveal that BELL-handoff significantly decreases both mobile devices' energy consumption and latency during handoffs, compared with the commercial WLAN mobility management service. Furthermore, we conduct extensive simulations to evaluate APs' load and mobile devices' battery life within a large-scale deployment of BELL. Simulation results demonstrate that BELL not only balances the load among APs, but also prolongs the battery life of mobile devices.**

I. Introduction

The increasing popularity of mobile devices (e.g., smart phones and tablets) has boosted the expansion of demands on myriad mobile applications, especially the large data stream mobile applications (e.g., augmented reality gaming and object recognition). Wireless local area networks (WLANs), which can provide mobile users (MUs) with high data rate wireless connections, have been a top choice for MUs. However, how to make an MU, on the one hand, obtains stable wireless services with its frequent mobility and the resource contention from other MUs, and on the other hand, survives longer with its limited battery capacity, are the most critical challenges that MUs within WLANs are facing.

Therefore, due to the limited range and capacity of an individual access point (AP), WLANs offer various management services to help associated MUs maintain stable wireless services. Over the past decades, there are many papers on WLAN mobility management, aiming to help MUs cross cell boundaries without service disruptions. In addition, a lot of load balancing mechanisms are designed in order to provide fair throughput for the attached MUs. Therefore, these management services can help MUs to obtain stable network throughput within WLANs. However, since in WLANs, APs are managed autonomously and independently, MUs need to consistently update the knowledge of neighboring APs, e.g., the received signal strength (RSS), data rate, and load, by sensing all available channels. For instance, active scanning is a major sensing approach in WLANs, where MUs actively broadcast probe packets on each channel. Therefore, in most cases, obtaining the a-priori knowledge of neighboring APs is at the expense of consuming a lot of energy on mobile devices.

This work was supported in part by the US National Science Foundation (NSF) under Grant No.1343355, 1718666, and 1731675

A recent study [1] shows that 23.4% of the average daily energy drain of mobile devices is for Wi-Fi beacons, Wi-Fi scanning, and cellular paging, which is the second largest portion of daily energy consumption. Furthermore, Wi-Fi scanning drains 28.1% of the total wireless energy consumption, and this number increases when MUs maintain a longer Wi-Fi connecting time. In other words, although more frequent scanning may obtain more updated information for conducting management services, it is at the cost of shortening mobile devices' battery life. Therefore, there exists a trade-off between the performance of management services and the battery life of mobile devices. Unfortunately, most of existing management approaches choose to sacrifice the energy efficiency of MUs, and thus, make mobile devices in energy crisis.

Existing work on WLAN management services and energy efficiency of mobile devices has the following issues. First, very few existing mobility management mechanisms for WLANs consider the energy efficiency of mobile devices. Early work on mobility management in WLANs mostly focuses on reducing the handoff latency (including the latency on scanning, authentication, and re-association). One category of these mechanisms is to decrease the scanning latency by reducing the number of channels to scan during a handoff based on different channel selection strategies [2]–[8]. However, the performance of these mobility management approaches is heavily limited by the channel selection strategy and user experience might be affected due to an insufficient scan. Thus, in order to avoid insufficient scan, mobility prediction mechanisms [8] are proposed to obtain a relatively precise channel scanning. Some papers, such as [6], propose to conduct frequent periodic scanning and caching before the mobile device triggers a handoff. Although these approaches achieve efficient scanning and low handoff latency, performing the mobility prediction, frequent periodic scanning, and caching consumes a large amount of additional energy of mobile devices, especially for MUs with high mobility. In other words, the improvement on mobility management performance is at

978-1-5386-4281-8/18/$31.00 ©2018 IEEE

the expense of the mobile device energy consumption. Another category of mobility management mechanisms is to reduce the time spent on scanning each channel. SyncScan [9] proposes that all the APs operating in the same channel broadcast their beacons at the same time and thus, mobile devices can scan a channel right before that time, which simplifies the scanning process and reduces the overall handoff latency. However, this protocol is hard to implement in practical scenarios, since APs that have overlapping coverage areas have to contend the channel for broadcasting their beacons, which indicates that mobile devices may miss the beacon broadcast from their associated APs because of APs' channel contention. Mobility management mechanisms in [10], [11] propose turning the MU mobility management into virtual AP migration inside the infrastructure network. However, this mechanism is also hard to popularize in practical scenarios, since it needs to create a virtual AP for each MU, which requires high storage and computing capacity of APs' hardware. In general, few of the prior mobility management papers propose a practical mechanism that consider both handoff latency and energy efficiency of mobile devices. Even worse, some may even increase the energy consumption of mobile devices in order to reduce the handoff latency. Therefore, proposing a mobility management mechanism that can reduce not only the handoff latency but also the energy consumption of mobile devices is highly desired.

Second, an early study [12] has shown that the traffic load is often unevenly distributed among the APs in WLANs. To overcome this, various load-balancing mechanisms have been proposed over the past decade. These mechanisms can be roughly classified into two categories: user-controlled and network-controlled. In user-controlled mechanisms [13], [14], MUs learn APs' load or measure the transmitting data rate and bit error rate to determine the AP to associate to. Network-controlled mechanisms [15]–[17], on the other hand, employ a network-side server or AP to manage the load of APs. Although the above work can balance the load of WLANs, MUs might suffer a connection disruption during the load transfer, because of the long transfer latency. Furthermore, user-controlled mechanisms require MUs to make frequent network sensing, which is a significant energy-cost. Therefore, a user-friendly (i.e. low transfer latency and low energy cost) load-balancing mechanism in WLANs is necessarily desired.

At last, although some papers analyze the energy consumption of mobile devices in WLANs, very few of them measure and investigate the energy consumption originated from management services process. Some papers [18]–[20] develop energy consumption models to analyze the energy cost of mobile devices in WLANs. Mostly, these models have a finite number of states, e.g., {active, idle} [19] and {transmission, reception, idle} [20]. However, a common approach followed in most of these papers is to model the energy consumption of the network interface card (NIC) by using data sheet parameters, and consider this as a fixed value to account for the non-wireless power consumption of the device. Therefore, these energy consumption models fail to capture crucial aspects of how energy is consumed in real-world devices, and thus, their use might bias conclusions. Some other papers [21]–[23] propose several assistant strategies to help mobile devices save energy in daily use. For instance, [22] proposes mobile devices can save energy using a radio interface selection strategy, e.g., using WLAN or 4G in a certain scenario would be more energy-efficient for communications. [23] proposes to reduce mobile devices' energy consumption by enabling MUs to access the WLAN infrastructure via low-power Bluetooth. However, none of the above work deeply investigates how the practical process of management services consume mobile devices' energy.

In this paper, we propose a novel load-**B**alanced, **E**nergy-efficient, and **L**ow-**L**atency WLAN system, named **BELL**, which offers energy-efficient mobility management and load balancing services for MUs. The main contributions of this work are as follows:

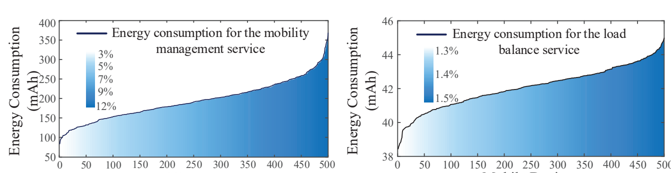
1) To the best of our knowledge, BELL is the first work that investigates MUs' energy efficiency in mobility management and load-balancing services in WLANs.
2) We propose BELL-handoff, an energy-efficient mobility management protocol. In addition, based on BELL-handoff, we develop a user-friendly load-balancing algorithm, BELL-2M, and prove that BELL-2M can always find the optimal load-balancing solution.
3) We implement BELL-handoff in commercial Wi-Fi adapters and conduct an experimental evaluation which shows that it can provide significant improvement in both mobile device handoff energy efficiency and latency, as compared to commercial Wi-Fi products.
4) We also conduct extensive simulations to evaluate APs' load and mobile devices' battery life within a large-scale deployment of BELL. Simulation results validate that BELL not only balances the load among APs but also prolongs the battery life of mobile devices.

The rest of this paper is organized as follows. In Section II, we show how WLAN management services consume mobile devices' energy. In Section III, we describe our proposed BELL system. Experimental and simulation results are shown in Section IV, followed by the conclusions in Section V.

II. Motivation

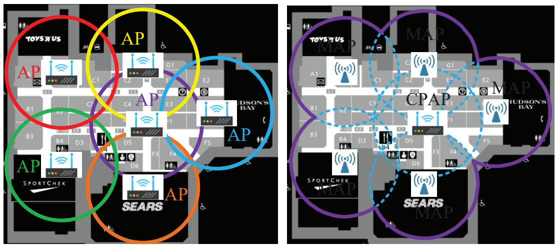
In this section, we show how existing WLAN mobility management service and load balancing service consume mobile devices' energy.

We have conducted an energy consumption measurement study of roaming mobile devices, including scanning and



(a) Mobility management service (b) Load balancing service

Fig. 1. Mobile devices' energy consumption for WLAN management services.



(a) Conventional WLAN deployment (b) Proposed BELL deployment

Fig. 2. Comparison of WLAN deployments.

handoff, using Motorola Nexus 6p smart phones. The measurement shows that performing a successful handoff with one scanning attempt consumes approximate 1.5 Joule. This energy consumption may dramatically increase if the Nexus 6p attempts to scan channels for more than once, i.e., the first scanning attempt fails to capture an appropriate AP to connect to. We then utilize this measurement result to conduct a trace-driven simulation, where 500 MUs walk in an area with 36 APs deployed for 5.5 hours. Each mobile device is attached with a 3000mAh battery and follows the Random Waypoint model. Fig.1(a) and Fig.1(b) depict the energy consumption of performing mobility management services and load balancing services, respectively. Each device's total energy consumption is sorted from low to high as shown in the X-axis. From the figure we can see that most devices consume approximate 8% and 1.4% of the total battery life for performing mobility management and load balancing services, respectively. Some mobile devices with high mobility consume a whopping 13.4% of the total battery life, 400mAh, for these two WLAN management services, which is equivalent to the battery drain for watching video around 1 hour. Therefore, reducing the energy consumption of management services is crucial and will help MUs survive longer within the WLAN environment.

Ultimately, the reason for MUs consuming a lot of energy for mobility management service and the load balancing service is performing frequent and periodic scans. Since the signal coverage and capacity of a single AP is considerably limited, MUs have to frequently and periodically update their knowledge of surrounding APs and switch between different APs to maintain their wireless connections. For example, consider that an MU is hanging out in a shopping mall, and a WLAN is deployed as shown in Fig. 2(a). A mobility management service is triggered when the MU walks across the signal boundary of APs, which is depicted as colored circles in Fig. 2(a). When the MU moves, for instance, walking from the TOYS R US store (up left corner) to the SEARS store (down right corner), the MU has to perform at least two handoffs, including channel scans, to maintain its wireless connection. However, in practical scenarios, users' path may not just follow a straight line. They may erratically visit the stores they are interested in one by one. Meanwhile, MUs have to keep periodic scanning and perform load balancing services (i.e. handoff from the AP with heavy load to the AP with light load) when their associated APs' load are heavy. Hence, MUs might perform much more number of scans and handoff attempts, which may consume a lot of energy of mobile devices and badly affect user experience because of the high handoff latency and load transfer delay.

To overcome the above problem, we propose BELL that virtually extends the signal coverage of a single AP. As shown in Fig. 2(b), the signal coverage of the AP located in the center is extended by the support of other APs around it. These APs are called *mirror access points* (MAPs) for the central one. They have the same SSID and MAC address with the center AP, named *copied physical access point* (CPAP). APs that generate these MAPs are called host physical access points. MAPs and the CPAP compose the proposed novel WLAN system, BELL. Since an MU would not filter out beacons that are generated from APs obtaining the same SSID and MAC address with its associated AP, the MU can receive beacons broadcast by MAPs and the CPAP in a BELL without scanning. In other words, MUs in a BELL would consider they are always in a single AP. Based on BELL, we propose an energy-efficient mobility management service, *BELL-handoff*, and a user-friendly load balancing service, *BELL-2M*.

III. The Proposed BELL System

In this section, we describe our proposed two WLAN management services, mobility management service, BELL-handoff, and load balancing service, BELL-2M, in detail.

A. BELL-handoff Mobility Management Service

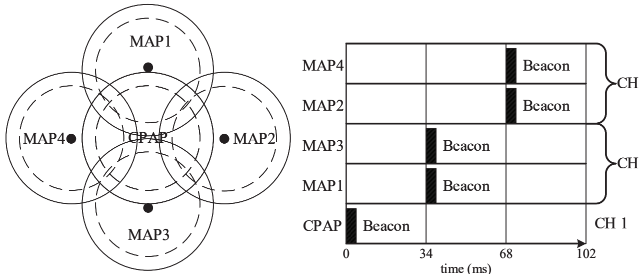
1) AP-side Protocol: At the heart of the proposed BELL-handoff is to create an evenly spaced periodic schedule of beacon periods for MAPs and the CPAP. The only assumption for the proposed AP-side protocol is that neighboring APs in a single BELL have overlapping areas, which is practical in AP-dense environments, and we do not have any other constraints on the AP deployment. Consider a BELL system that has $(N+1)$ number of APs (i.e., N MAP and a CPAP). The CPAP is operating in channel v, and its beacon is broadcast at time $\phi_{cpap} \in [0, T]$, periodically, where T is the beacon interval. Each MAP has an index number $i \in \{1, \ldots, N\}$ which is allotted clockwise. Let MAPs with an odd index operate in channel p and MAPs with an even index operate in channel q. Channel v, p, q are three non-overlapping channels (e.g., $\{v, p, q\} = \{1, 6, 11\}$ in the 2.4 GHz band). If N is even, then

$$\phi_i = \begin{cases} \phi_{cpap} + \frac{T}{\chi}, & \text{if } i = 2n+1 \\ \phi_{cpap} + (\chi - 1) \cdot \frac{T}{\chi}, & \text{otherwise} \end{cases}$$

where $n \in \{0, \ldots, \frac{N-1}{2}\}$, ϕ_i denotes the beacon broadcasting time of MAP i, and χ denotes the number of segments in the beacon interval. If N is odd,

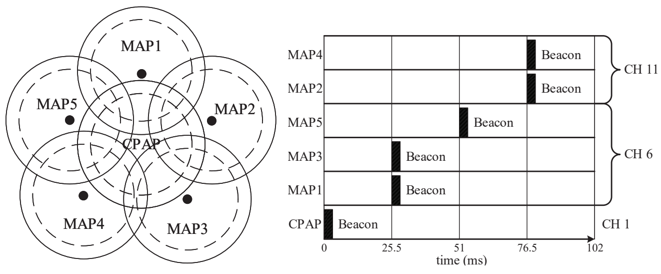
$$\phi_i = \begin{cases} \phi_{cpap} + \frac{T}{\chi}, & \text{if } i = 2n+1 \\ \phi_{cpap} + (\chi - 1) \cdot \frac{T}{\chi}, & \text{if } i = N \\ \phi_{cpap} + (\chi - 2) \cdot \frac{T}{\chi}, & \text{otherwise} \end{cases}$$

where $n \in \{0, \ldots, \frac{N-2}{2}\}$. Furthermore, when N is even, $\chi = 3$ (i.e., the total number of operating channels in the BELL), otherwise $\chi = 4$.



(a) BELL deployment (b) Beacon broadcast schedule

Fig. 3. Example of a BELL with even number of MAPs.



(a) BELL deployment (b) Beacon broadcast schedule

Fig. 4. Example of a BELL with odd number of MAPs.

Example 1. Consider a BELL system with even number of MAPs (e.g., $N = 4$ in this example) operating in the 2.4 GHz band, as shown in Fig. 3(a). APs' physical signal transmission range is depicted as the solid circles, and BELL-handoff triggering range is depicted as the dashed circles. Assume the CPAP, MAP $\{1, 3\}$, and MAP$\{2, 4\}$ are operating in channel 1, 6, and 11, respectively. $\chi = 3$, $\phi_{cpap} = 0$. All the APs in the BELL are set for $T = 102ms$. Therefore, their beacon broadcasting time is set to be $\{34, 68\}ms$, as shown in Fig. 3(b). Furthermore, there would be no collision for different MAPs broadcasting beacons at the same time, since they do not have overlapping coverage area.

Example 2. Consider a BELL system with odd number of MAPs (e.g., $N = 5$ in this example) operating in the 2.4 GHz band, as shown in Fig. 4(a). Assume that the CPAP, MAP $\{1, 3, 5\}$, and MAP$\{2, 4\}$ are operating in channel 1, 6, and 11, respectively. $\chi = 4$, $\phi_{cpap} = 0$, and $T = 102ms$. Therefore, their beacon broadcasting time is set to be $\{25.5, 51, 76.5\}ms$, as shown in Fig. 4(b).

2) User-side Protocol: MUs in BELL perform our proposed energy-efficient BELL-handoff. Consider an MU u currently located in MAP i ($1 < i < N$) and a BELL-handoff is triggered since the received signal strength (RSS) at the MU is below a threshold γ. The possible candidate AP set for MU u to connect to is $\{$MAP $i+1$, MAP $i-1$, CPAP$\}$. Therefore, MU u only needs to switch to the corresponding channels one by one, and waits for the beacon at each beacon broadcasting time according to the BELL beacon schedule. MU u stops this channel switching when successfully receives a beacon frame. For example, consider a BELL system as deployed in Fig. 3. Assume a BELL-handoff is triggered when MU u is in MAP 3 at time t_0. It first switches its channel to CH 11 and completes its BELL-handoff if it receives a beacon frame at $(t_0+34)ms$. Otherwise, it switches its channel again to CH 1 and waits for receiving the beacon broadcast at $(t_0+68)ms$. Therefore, MUs can successfully roam in BELL without performing scanning, achieving energy-efficient handoffs. Note that although after a BELL-handoff, an MU may be handed off to an MAP with RSSI below γ, it can trigger another BELL-handoff soon and connect to an MAP with better RSSI.

3) Implementation Challenges: There exist some issues when implementing our presented BELL system both in APs and in user devices.

Challenge 1. *To have an accurate beacon schedule over long time scales, it is necessary to overcome the clock drift that exists in each AP.*

We address this issue by proposing an AP *clock synchronization* (CS) protocol. In the proposed CS protocol, all the MAPs in a BELL adjust their clock every ψ, by synchronizing its local Timing Synchronization Function (TSF) timer with a common clock which is the local TSF timer of the CPAP. ψ is an integral multiple of the beacon interval T. Assume that the CPAP always broadcast its beacons first in every beacon interval. In addition, at time σ, MAP i and the CPAP have a local TSF value of $t_i(\sigma)$ and $t_{cpap}(\sigma)$, respectively. Therefore, if N is even, then

$$t^i_{offset}(\sigma) = \begin{cases} |t_i(\sigma) - t_{cpap}(\sigma)| - \frac{T}{\chi}, & \text{if } i = 2n+1 \\ |t_i(\sigma) - t_{cpap}(\sigma)| - \frac{(\chi-1)\cdot T}{\chi}, & \text{otherwise} \end{cases}$$

where $n \in \{0, \ldots, \frac{N-1}{2}\}$, $t^i_{offset}(\sigma)$ denotes the timing offset value for MAP i at time σ. If N is odd, then

$$t^i_{offset}(\sigma) = \begin{cases} |t_i(\sigma) - t_{cpap}(\sigma)| - \frac{T}{\chi}, & \text{if } i = 2n+1 \\ |t_i(\sigma) - t_{cpap}(\sigma)| - \frac{(\chi-1)\cdot T}{\chi}, & \text{if } i = N \\ |t_i(\sigma) - t_{cpap}(\sigma)| - \frac{(\chi-2)\cdot T}{\chi}, & \text{otherwise} \end{cases}$$

where $n \in \{0, \ldots, \frac{N-2}{2}\}$. Then,

$$\bar{t_i}(\sigma) = t_i(\sigma) - t^i_{offset}(\sigma), \tag{1}$$

where $\bar{t_i}(\sigma)$ denotes the corrected TSF for MAP i at time σ.

Challenge 2. *How does an MU acquire enough MAPs' information to achieve a successful BELL-handoff?*

In the proposed BELL system, MUs do not need to know the index of a particular MAP. Instead, acquiring BELL's beacon broadcasting schedule and its corresponding channels is enough for performing a successful BELL-handoff. For example, consider a BELL as deployed in Fig. 3 and an MU is in MAP 3. The MU only knows it is currently in channel 6. After a BELL-handoff is triggered, the MU just jump to channel 11 and channel 1 in order according to the beacon broadcasting schedule. Therefore, we only ask each MU acquiring a beacon broadcasting schedule while joining BELL, which is simple enough to implement.

Furthermore, the deployment of the BELL presented above is based on an assumption that, MAPs with an even index or an odd index can operate in the same channel and have the same beacon broadcast time, only if they do not have overlapping coverage area. For example, in Fig. 3, MAP1 and MAP3 operating in the same channel and having the same beacon broadcast time do not have any overlapping coverage area. However, if the density of MAPs in a BELL is increased, two neighboring MAPs with an even index or an odd index may have overlapping coverage area. To avoid the interference, their beacons need to broadcast at different time or they need to operate in different channels. Therefore, there exists a limitation of the maximum number of MAPs in a BELL: the minimal gap in time between two beacons that broadcast by different MAPs has to be larger than the MU's channel switching delay, i.e., $\frac{T}{\chi} > t_{cs}$, where t_{cs} is the MU's channel switching delay.

4) Analysis:

1) Energy efficiency: as presented in Section III-A, MUs roaming in BELL do not perform scanning, which indicates that they do not need to send or receive probe request and response for the purpose of scanning. Therefore, the proposed BELL provides a more energy-efficient mobility management service than conventional WLAN (C-WLAN).
2) Handoff latency: as presented in Section III-A, the handoff latency in BELL is equal to the delay of waiting for the beacon, plus the delay for authentication and re-association which is the same as in traditional handoff. Excluding authentication and re-association delay,

$$PLS = \begin{cases} \{\frac{T}{\chi}, (\chi-1)\cdot\frac{T}{\chi}\}, & \text{if } N \text{ is even} \\ \{\frac{T}{\chi}, (\chi-2)\cdot\frac{T}{\chi}, (\chi-1)\cdot\frac{T}{\chi}\}, & \text{otherwise} \end{cases}$$

where PLS is the the possible handoff latency set. In our cases, T is set to be $102ms$. Thus, ideally, the maximum handoff latency, excluding authentication and re-association delay, is $68ms$, if N is even, and it is $76.5ms$, if N is odd.

5) Mutiple BELL Zones Scenario: In the mutiple BELLs scenario [24], several BELL WLANs coexist in a certain area. Each AP can be a CPAP, which indicates that a host physical AP would generate multiple different MAPs in it, as illustrated in Fig. 5. The number of MAPs generated in a host physical AP is equal to the number of MAPs in this BELL. Different BELL systems can have the same channel set. In addition, MUs roaming between different BELL WLANs perform conventional handoffs.

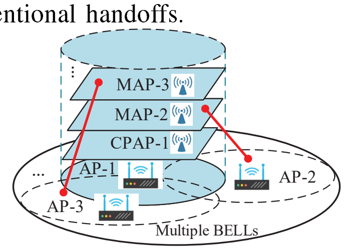


Fig. 5. Multiple BELL zones.

B. BELL-2M Load Balancing Service

1) Problem Statement: Balancing the load on APs is a primary way for MUs obtaining a fair service. However, the definition of the load of an AP is fuzzy. Thus, determining an appropriate load definition is necessary. *Intuitively, the load of an AP needs to reflect its inability to satisfy the requirements of its associated users and as such it should be inversely proportional to the average bandwidth that they use.* We present our load definition in WLANs which captures the above summary.

Assume that the proposed BELL system has a set of MAPs, denoted by A. N denotes the number of MAPs in the system. The MAPs having the maximal load are called the load-heaviest MAPs and their load is denoted as y_{max}. In addition, without loss of generality, in our BELL-2M, we only reserve one MAP as the load-heaviest one at each iteration. Let U denote the set of all MUs in the BELL coverage area and M denote the number of MUs.

Definition 1 (The Load of an MAP)**.** *Consider an MAP $a \in A$, and let U_a be the set of MUs associated with MAP a. The load of MAP a, denoted by y_a, is the aggregate period of time that takes MAP a to provide a unit of traffic volume to all associated MUs $u \in U_a$. Thus,*

$$y_{a\in A}^{U_a} = \sum_{u\in U_a}(\frac{1}{r_{a,u}}), \tag{2}$$

where $r_{a,u}$ is the wireless link bit rate between MAP a and MU u.

Definition 2 (BELL-2M Load-Balanced Vector)**.** *We define the BELL load vector as $\vec{Y} = \{y_1^{U_1}, \ldots, y_N^{U_N}\}$, which is BELL-2M balanced when the load of any MAP can not be reduced only if increasing another MAP load with the same or higher load.*

We can show that the problem of finding a BELL-2M load-balanced vector is NP-hard, by proving that even the simplest scenario, e.g., only two MAPs in the BELL, is NP-hard.

Proof. Consider a case with only two MAPs. Each MU $u \in U$ can be covered by both MAPs. Therefore, to obtain the BELL-2M load-balanced vector $\vec{Y}$ equals to find a subset $U' \subseteq U$ that satisfies

$$y_1^{U'} = y_2^{U-U'}. \tag{3}$$

By restricting the problem to $\frac{1}{r_{1,u}} = \frac{1}{r_{2,u}}$, the problem can be a reduction from the partition problem. *The partition problem is the task of deciding whether a given multiset S of positive integers can be partitioned into two subsets S_1 and S_2 such that the sum of the numbers in S_1 equals the sum of the numbers in S_2*, which is a known NP-hard problem. □

BELL-2M aims to iteratively find an optimal BELL-2M load-balanced vector $\vec{Y}$ in polynomial time.

2) BELL-2M Algorithm: The basic idea is that, at each iteration, g, we reduce the load of the load-heaviest MAP a at this stage by transferring some of its associated MUs to its neighboring MAPs. Furthermore, MUs with lower RSSI have a higher priority to be transfered. In other words, we define the RSSI vector $\overrightarrow{UR_a} = \{\mu_1, \ldots, \mu_m\}$ to be an m-tuple consisting of the RSSI of each MU in MAP a sorted in an increasing order, where m is the total number of MUs in MAP a. MAP a always first transfers the MU with the lowest order in $\overrightarrow{UR_a}$.

However, there are two algorithmic challenges for the above problem:

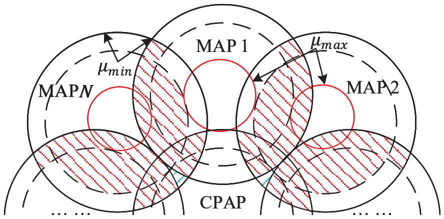


Fig. 6. Only MUs with RSSI in $[\mu_{min}, \mu_{max}]$ might have chance to be transferred.

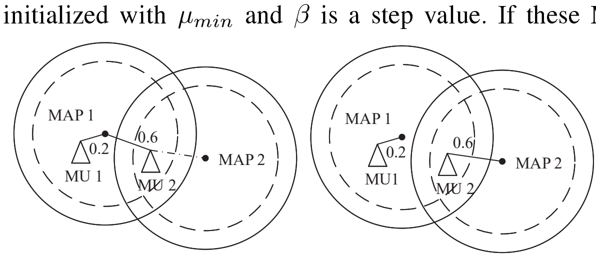
Challenge 3. *Not every MU in MAP a is covered by another neighboring MAP. In BELL, as shown in Fig. 6, MUs located within the blue shadow area have no chance to be successfully transferred to neighboring MAPs.*

Challenge 4. *The Ping-Pong effect of load-balancing decision.*

Example 3. Consider a BELL with two MAPs, as shown in Fig. 7, denoted as v_1 and v_2, and two MUs u_1 and u_2. u_1 can only connect to v_1 and it yields a load of 0.2. u_2 can connect to both v_1 and v_2 and yields a load of 0.6 on its connected MAP. Both u_1 and u_2 first connect to v_1, and the RSSI vector of v_1 is $\overrightarrow{UR_{v_1}} = \{\mu_{u_2}, \mu_{u_1}\}$. As shown in Fig. 7(a), the loads of v_1 and v_2 are 0.8 and 0, respectively. To balance the load, a greedy algorithm first lets v_1 transfer u_2 to v_2. Now the loads of the two MAPs are 0.2 and 0.6, respectively, as depicted in Fig. 7(b). Then, the algorithm lets v_2 reduce its load, since it is the load-heaviest MAP at this stage. Thus, v_2 transfers u_2 back to v_1, and the algorithm repeats the above indefinite iterations, which obviously can not obtain an optimal solution.

Our BELL-2M algorithm resolves the above two challenges by, firstly, introducing a maximal transfer RSSI μ_{max} and a minimal transfer RSSI μ_{min}. As shown in Fig. 6, we ask the MU with RSSI larger than μ_{max} (those MUs only covered by its associated MAP) not to be transferred. On the other hand, let μ_{min} be the RSSI corresponding to the MAP's transmission range. Thus, only MUs with RSSI in the range of $[\mu_{min}, \mu_{max}]$ might have chances to be triggered with a load transfer. Secondly, to overcome Challenge 4, we define a set of fixed MAPs, D, whose loads have already been determined by previous iterations. At the beginning, the set D is empty, and after each iteration, a new MAP is added to it, until $D = A$. Therefore, BELL-2M only searches the set of the non-fixed MAPs, $\{A - D\}$, for the load-heaviest MAP at each stage.

The proposed BELL-2M load-balancing algorithm is shown in Algorithm 1. At each iteration, the algorithm first finds the load-heaviest MAP α_0 and preserves its load y_{max} in θ. Then MAP α_0 checks MUs from its farthest possible MUs whose RSSI are in the range of $[K_{\alpha_0}, K_{\alpha_0} + \beta]$, where K_{α_0} is initialized with μ_{min} and β is a step value. If these MUs



(a) Original MU-MAP connection (b) MU-MAP connection after an iteration

Fig. 7. Example of the Ping-Pong effect of a simple greedy algorithm.

Algorithm 1: BELL-2M Load Balancing Algorithm

$D = \varnothing$
while $D \neq A$ **do**
  /* Find and record the load-heaviest MAP α_0 and its load y_{α_0} */
  $\alpha = \alpha_0 \ s.t. \ y_{\alpha_0} = \max_{\alpha_0 \in \{A-D\}} \{y_{\alpha_0}\}$
  $\theta = y_{max} = y_{\alpha_0}$
  $K_{\alpha_0} = \mu_{min}$
  while $K_{\alpha_0} < \mu_{max}$ **do**
    $c =$ the number of MUs with RSSI in $[K_{\alpha_0}, K_{\alpha_0} + \beta]$
    $K_{\alpha_0} = K_{\alpha_0} + \beta$
    /* Check if exist MUs with RSSI in this range */
    if $c > 0$ **then**
      for $i = 1$ **to** c **do**
        MUi performs BELL-handoff
        /* Find and record the load-heaviest MAP α_0 and its load y_{α_0} */
        $\alpha_0 \ s.t. \ y_{\alpha_0} = \max_{\alpha_0 \in \{A-D\}} \{y_{\alpha_0}\}$
        $y_{max} = y_{\alpha_0}$
        /* Check if fixed MAPs' load was not increased and a better load vector was found */
        if $(\nexists \alpha_0 \in D \ s.t. \ y_{\alpha_0} > \bar{y_{\alpha_0}}) \wedge (y_{max} < \theta)$ **then**
          $\theta = y_{max}$
          $\alpha = \alpha_0$
  $D = D \cup \{\alpha\}$
  $\bar{y_\alpha} = y_\alpha$

exist, MAP α_0 will evaluate the load transfer result of them one by one: i.e., if after transferring an MU, none of the fixed MAPs' load increases, and the new load-heaviest MAP's load is smaller than that of the previous stage, the checked MU will be confirmed to transfer. Otherwise, the MU will not be transferred. After all MUs are checked, K_{α_0} is incremented by β in each iteration. The evaluation stops if $K_{\alpha_0} = \mu_{max}$. Then, MAP α_0 joins the fixed MAP set as well as its load value. Finally, the algorithm stops when $D = A$, which means that all the MAPs' load has been fixed.

We now prove that the proposed algorithm always finds a BELL-2M load-balanced vector for BELL.

Proof. Each iteration starts with the load-heaviest MAP with $K_{\alpha_0} = \mu_{min}$ and stops when $K_{\alpha_0} = \mu_{max}$. MUs are transferred only if y_{max} is decreased after this transfer. Therefore, at the end of the gth iteration, y_{max} is not larger than its value at the beginning of the gth iteration. The corresponding MAP is added in the fixed MAP set D after the gth iteration. Since the definition of the load-heaviest MAP is the non-fixed MAP that has the maximal load at this stage, y_{max} at the $(g+1)$th iteration is smaller than that at the gth iteration. Also, MUs can only be transferred to the non-fixed MAPs, which means that the load of MAPs in D is not affected in the $(g+1)$th iteration. Therefore, the algorithm keeps reducing y_{max} and stops at an optimal solution. The algorithm stops when $D = A$, in other words, all the MAPs become load fixed MAPs. Therefore, y_{max} cannot be reduced further. According to Definition 2, the BELL-2M load-balanced vector is found. □

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed BELL. We first implement our proposed BELL-handoff mobility management service on a testbed and demonstrate MUs' roaming performance improvement in BELL, including handoff energy consumption and latency, as compared to the conventional WLANs (C-WLANs). Since implementing a large-scale deployment of MAPs is very costly, we conduct extensive simulations to evaluate the performance of BELL-2M and mobile devices' battery life within a large-scale deployment of BELL.

A. Testbed and Experimental Setup

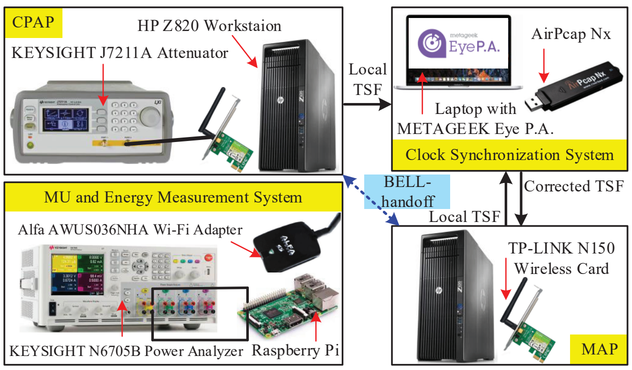


Fig. 8. Testbed hardware architecture.

Our testbed is composed of three subsystems, MU and energy measurement system, BELL including a CPAP and an MAP, and AP clock synchronization system. Firstly, in the MU and energy measurement system, we use KEYSIGHT N6705B DC Power Analyzer for powering and measuring the energy consumption of a Raspberry Pi Model 3 connected with an ALFA AWUS036NHA USB Wi-Fi adapter, which is conducted as a test MU. And the Raspberry Pi runs Ubuntu MATE 16.04.2. The Wi-Fi adapter is driven by an open-source hardware driver, ath9k, and its MAC functionality is handled by the protocol driver mac80211, which is where we implement BELL-handoff. We update our modified mac80211 to the Linux kernel which is then embedded into the test Raspberry Pi. Secondly, the tested BELL is setup with a CPAP and an MAP. These two are built in two workstations with the Linux operating system. Both workstations are connected with a TP-LINK N150 PCI-Express wireless card driven by ath9k and mac80211. Then, we use hostapd in userspace to set the configuration of these two APs. KEYSIGHT J7211A attenuator is used for emulating the mobility of the MU, e.g., decreasing the transmission signal strength of the CPAP emulates the case when an MU gradually moves away from the CPAP. Furthermore, to implement our AP's CS protocol, we modify ath9k by adding a function interface to rewrite the

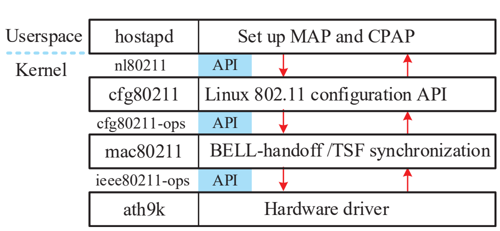


Fig. 9. Testbed driver software structure.

register that stores the dynamic local TSF of APs. Lastly, we use Airpcap Nx and its client software, Metageek Eye P.A., for capturing control frames at the MAC layer and extracting local TSF values of the CPAP and the MAP to conduct the clock synchronization. The testbed hardware architecture and driver software structure are shown in Fig. 8 and Fig. 9, respectively.

We set the beacon interval of APs in BELL to be $102ms$. Thus, the beacon broadcast time of the CPAP and the MAP are 0 and $51ms$, respectively. Both APs operate in the 2.4 GHz band. The CPAP operates in channel 1 and the MAP operates in channel 6. Furthermore, the test Raspberry Pi is powered by 5.2V and its BELL-handoff threshold is set to be -80dB.

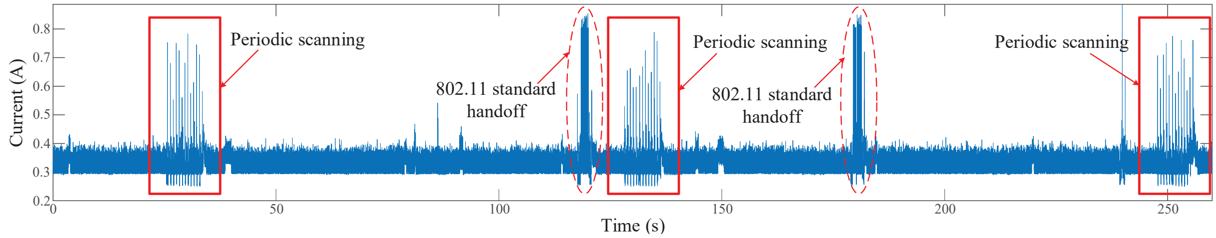
B. Mobility Management Service Performance

1) Energy Efficiency: We plot experimental results in Fig. 10. The measurement current of the test Raspberry Pi without connecting any peripherals is approximately fixed at 0.21A. After connecting a USB Wi-Fi adapter, the Raspberry Pi's current is increased to around 0.3A, as shown in Fig. 10(a). This indicates that the Wi-Fi adapter consumes an approximately fixed current of 0.09A in idle state. Fig. 10(a) shows the current measurement result of the Raspberry Pi within 250sec. We find that in approximately every 120sec, the mac80211 driver will trigger a periodic full channel scanning, which lasts around 9sec, as shown in Fig. 10(b). Each spike represents one scan on a channel. Since the MU still has a good-quality connection with its current associated AP, it switches back to the original channel to maintain its connection after scanning each channel. Fig. 10(c) and Fig. 10(d) compare the current of performing the 802.11 standard handoff within C-WLAN and the proposed BELL-handoff within BELL. Obviously, both the current value and duration of performing the proposed BELL-handoff is much smaller than that of performing the 802.11 standard handoff, which indicates that the proposed BELL provides energy-efficient mobility management service for MUs.

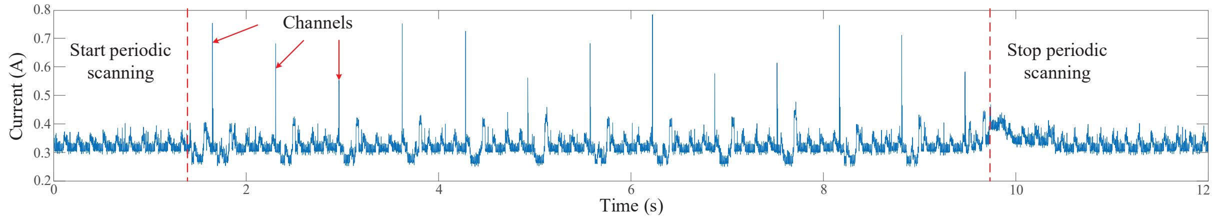
TABLE I
HANDOFF ENERGY CONSUMPTION RESULTS

	Overall	Breakdown	
C-WLAN (periodic scanning)	10.138J		
C-WLAN (802.11 standard handoff)	4.366J	Trigger	0.902J
		Scan	2.809J
		Auth and re-assoc	0.655J
BELL (BELL-handoff)	0.748J	Channel switching	0.038J
		Idle	0.050J
		Auth and re-assoc	0.660J

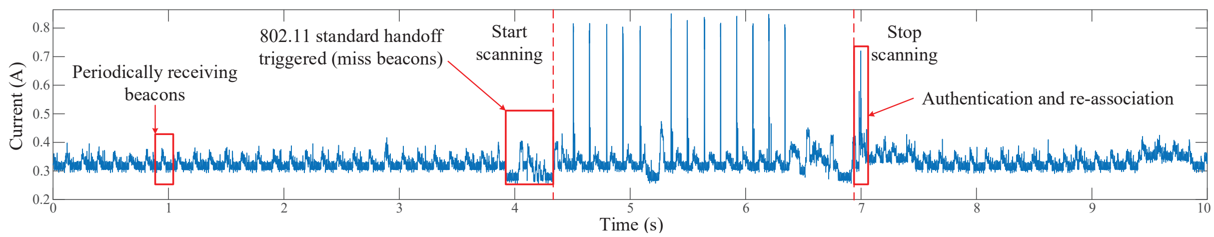
The calculated energy consumption during handoffs is depicted in Table I. The proposed BELL-handoff decreases the energy consumption by around 82.9% as compared with the 802.11 standard handoff protocol. Furthermore, we find that, in 802.11 standard handoff protocol, the handoff trigger is a number of continuously missed beacon frames instead of a fixed RSSI value. Therefore, it would consume an additional energy of 0.902J for triggering. The periodic scanning in 802.11 standard protocol consumes 10.138J energy. This would aggravate MUs' energy drain. However, in BELL, MUs perform neither periodic scanning nor scanning within



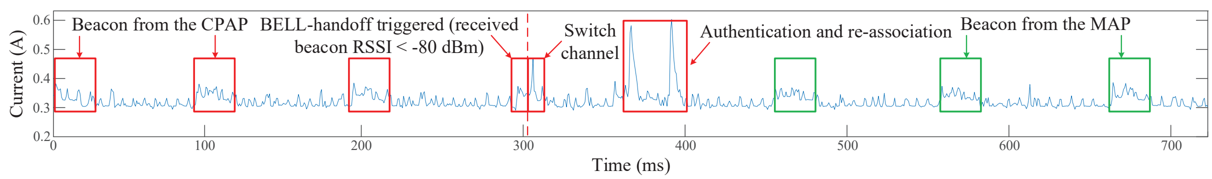
(a) Current of the test Raspberry Pi connecting a USB Wi-Fi adapter within C-WLAN



(b) Current of the test Raspberry Pi for performing a periodic full channel scanning within C-WLAN



(c) Current of the test Raspberry Pi for performing one 802.11 standard handoff within C-WLAN



(d) Current of the test Raspberry Pi for performing one BELL-handoff within BELL

Fig. 10. Measured current of MU.

mobility management services, which significantly reduces the handoff energy consumption of MUs.

2) Latency: The experimental results are shown in Table II. The proposed BELL-handoff decreases the overall handoff latency from 3.06sec to 75.6ms, as compared with C-WLANs. We can see that, in C-WLANs, the scanning latency, 2.62sec, is the largest portion of the total handoff latency. In addition, since the handoff trigger mechanism for the 802.11 standard handoff is to count the number of continuously missed beacons, the trigger delay, which is measured as 0.41sec, increases the total handoff latency. However, the delay of BELL-handoff is from three parts: the MU's channel switching delay, which is less than 1ms, the time for waiting for the new beacon, around 51ms, which depends on the beacon scheduling in BELL, and authentication and re-association delay, around 24ms. In total, the proposed BELL-handoff decreases the handoff latency by around 97.5%, as compared with the C-WLAN.

TABLE II
HANDOFF LATENCY RESULTS

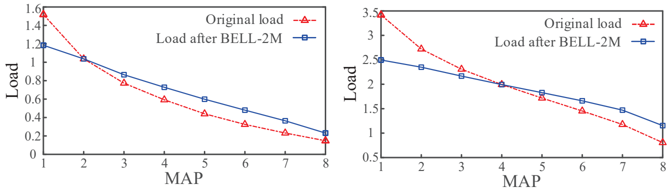
	Overall	Breakdown	
C-WLAN (802.11 standard handoff)	3.06sec	Trigger	0.41sec
		Scan	2.62sec
		Auth and Re-assoc	24ms
BELL (BELL-handoff)	75.6ms	Channel switching	$<$ 1ms
		Idle	51ms
		Auth and Re-assoc	24ms

C. Simulation Setup

The simulation setting is as follows. 9 APs are regularly distributed in a square are. The length of the simulation area is $600m$. Each AP has a circular transmission range, and its radius is $145m$. In BELL, the operating channel of the CPAP, MAPs with odd index, and MAPs with even index are 1, 6, and 11 in 2.4 GHz band, respectively. In C-WLAN, the operating channel of the APs are chosen randomly in 2.4 GHz band. To determine the bit rate of the connection between an MU and an AP, Signal-to-Noise Ratio (SNR) is computed and the bit rate is chosen accordingly. We set each AP's transmission power to be 20 dBm and the value of Δ to be 5. To simulate an indoor environment, we use the indoor path-loss model which is expressed as

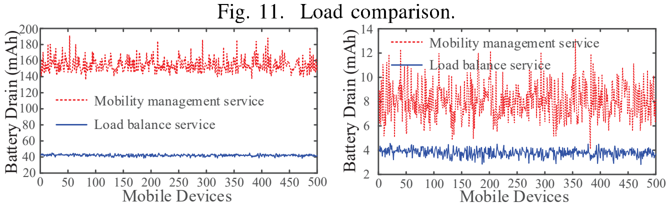
$$P_L = 20 \log_{10} f + 10n \log_{10} d - 28(dB), \tag{4}$$

where P_L is the RF signal propagation path-loss based on distance d between the AP and the MU, f is the carrier frequency in MHz, and n is the path-loss exponent. In our simulation, $n = 6$ and $f = 2400$ MHz. The background noise level is set to be -90 dBm. In the simulation of evaluating BELL-2M load balancing performance, the number of M MUs



(a) $M = 100$ (b) $M = 300$

Fig. 11. Load comparison.



(a) Battery drain within C-WLAN (b) Battery drain within BELL

Fig. 12. Battery drain comparison.

are randomly positioned in the simulation area and they are assumed to follow the saturated traffic model and quasi-static. In the simulation of evaluating the battery drain of mobile devices, 500 MUs follow the Random Waypoint model and walk in the simulation area for 5.5 hours.

D. Simulation Results

1) Load Balancing Performance: Fig. 11 shows the simulation results of the load balancing performance. The Y-axis represents the load of MAPs y_a and the X-axis represents MAP index. MAPs are sorted by their load values in a decreasing order. In Fig. 11(a) and Fig. 11(b), we obtain their load values by running simulations at least 500 times. In other words, each load value, y_a is obtained by averaging the 500 simulation load results. The red dotted line represents the original load of BELL, and the blue solid line represents the load of BELL after performing our BELL-2M load-balancing protocol. It is confirmed that BELL-2M can improve the load-balancing performance of BELL.

2) Mobile Devices' Battery Drain: Fig. 12 illustrates the simulation results of mobile devices' battery drain for WLAN management services within 5.5 hours. The Y-axis represents the battery drain of mobile devices and the X-axis represents MU index. Comparing Fig. 12(a) and Fig. 12(b), it is shown that MUs in BELL drain 94% and 90% less battery life for mobility management service and load balancing service, respectively, than MUs in C-WLAN.

V. Conclusion

In this paper, a novel WLAN system, BELL, is proposed to reduce mobile devices' energy consumption for performing management services. First, an energy-efficient mobility management service, BELL-handoff, is proposed under BELL deployment. We implemented BELL-handoff on a testbed and evaluated its performance. Experimental results show BELL-handoff decreases not only the handoff energy consumption but latency, as compared with C-WLAN. Then, we proposed a user-friendly load balancing service, BELL-2M. We conducted extensive simulations to evaluate BELL-2M performance and mobile devices' battery life within a large-scale deployment of BELL. Simulation results show that BELL-2M can balance APs' load in BELL and MUs in BELL drain less battery life than in C-WLAN.

References

[1] X. Chen, A. Jindal, N. Ding, Y. C. Hu, M. Gupta, and R. Vannithamby, "Smartphone background activities in the wild: Origin, energy drain, and optimization," in *Proc. ACM MobiCom*, 2015, pp. 40–52.

[2] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proc. ACM MobiWac*, 2004, pp. 19–26.

[3] M. Tuysuz and H. A. Mantar, "Minimizing communication interruptions using smart proactive channel scanning over IEEE 802.11 WLANs," *Wireless Personal Communications*, vol. 82, pp. 2249–2274, 2015.

[4] W. Zhao and J. Xie, "IMeX: intergateway cross-layer handoffs in internet-based infrastructure wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 1585–1600, 2012.

[5] J. Xie, "User independent paging scheme for mobile IP," *Wireless Networks*, vol. 12, pp. 145–158, 2006.

[6] S. Pack, H. Jung, T. Kwon, and Y. Choi, "Snc: a selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, pp. 39–49, 2005.

[7] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive scan: Fast handoff with smart triggers for 802.11 wireless LAN," in *Proc. IEEE INFOCOM*, 2007, pp. 749–757.

[8] W. Wanalertlak, B. Lee, C. Yu, M. Kim, S.-M. Park, and W.-T. Kim, "Behavior-based mobility prediction for seamless handoffs in mobile wireless networks," *Wireless Networks*, vol. 17, pp. 645–658, 2011.

[9] I. Ramani and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," in *Proc. IEEE INFOCOM*, 2005, pp. 675–684.

[10] Y. Grunenberger and F. Rousseau, "Virtual access points for transparent mobility in wireless LANs," in *Proc. IEEE WCNC*, 2010, pp. 1–6.

[11] M. E. Berezin, F. Rousseau, and A. Duda, "Multichannel virtual access points for seamless handoffs in ieee 802.11 wireless networks," in *Proc. IEEE VTC Spring*, 2011, pp. 1–5.

[12] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *Proc. ACM MobiSys*, 2003, pp. 303–316.

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

[14] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection," in *Proc. ACM MobiSys*, 2006, pp. 233–245.

[15] Y. Bejerano and S.-J. Han, "Cell breathing techniques for load balancing in wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 735–749, 2009.

[16] J. Xie and I. Howitt, "Multi-domain WLAN load balancing in WLAN/WPAN interference environments," *IEEE Transactions on Wireless Communications*, vol. 8, 2009.

[17] X. Chen, Y. Zhao, B. Peck, and D. Qiao, "Sap: Smart access point with seamless load balancing multiple interfaces," in *Proc. IEEE INFOCOM*, 2012, pp. 1458–1466.

[18] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in *Proc. ACM CoNEXT*, 2012, pp. 169–180.

[19] M. M. Carvalho, C. B. Margi, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Modeling energy consumption in single-hop IEEE 802.11 ad hoc networks," in *Proc. IEEE ICCCN*, 2004, pp. 367–372.

[20] X. Wang, J. Yin, and D. P. Agrawal, "Analysis and optimization of the energy efficiency in the 802.11 DCF," *Mobile Networks and Applications*, vol. 11, pp. 279–286, 2006.

[21] S.-L. Tsao and C.-H. Huang, "A survey of energy efficient MAC protocols for IEEE 802.11 WLAN," *Computer Communications*, vol. 34, pp. 54–67, 2011.

[22] M. F. Tuysuz, "An energy-efficient QoS-based network selection scheme over heterogeneous WLAN–3G networks," *Computer Networks*, vol. 75, pp. 113–133, 2014.

[23] J.-W. Yoo and K. H. Park, "A cooperative clustering protocol for energy saving of mobile devices with WLAN and Bluetooth interfaces," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 491–504, 2011.

[24] H. Wang, J. Xie, and T. Han, "V-handoff: A practical energy efficient handoff for 802.11 infrastructure networks," in *Proc. IEEE ICC*, 2017.