

Greening Residential WiFi Networks under Centralized Control

Vijay Sivaraman, *Member, IEEE*, John Matthews, Craig Russell, Syed Taha Ali and Arun Vishwanath *Member, IEEE*,



Abstract—Residential broadband gateways (comprising modem, router, and WiFi access point), though individually consuming only 5-10 Watts of power, are significant contributors to overall network energy consumption due to large deployment numbers. Moreover, home gateways are typically always on, so as to provide continuous online presence to household devices for VoIP, smart metering, security surveillance, medical monitoring, etc. A natural solution for reducing the energy consumption of home gateways is to leverage the overlap of WiFi networks common in urban environments and aggregate user traffic on to fewer gateways, thus putting the remaining to sleep. In this paper we propose, evaluate, and prototype an architecture that overcomes significant challenges in making this solution feasible at large-scale. We advocate a centralized approach, whereby a single authority coordinates the home gateways to maximize energy savings in a fair manner. Our solution can be implemented across heterogeneous ISPs, avoids client-side modifications (thus encompassing arbitrary user devices and operating systems), and permits explicit control of session migrations. We apply our solution to WiFi traces collected in a building with 30 access points and 25,000 client connections, and evaluate via simulation the trade-offs between energy savings, session disruptions, and fairness. We then prototype our system on commodity WiFi access points, test it in a two-storey building emulating 6 residences, and demonstrate radio energy reduction of over 60% with little impact on user experience.

Index Terms—Home WiFi networks, energy consumption, bandwidth aggregation, centralized control, system evaluation.

1 INTRODUCTION

The energy consumption of the ICT sector in general, and communication networks in particular, is receiving increasing attention due to its alarming growth rate [1]. It is also recognized that a majority of the power is consumed in the access network [2], due to the sheer volume of user premises equipment (typically a home gateway comprising a modem, router, and wireless access point) and their high per-bit energy consumption. Even at conservative estimates of 5 Watts per home gateway device (which is typically always on), in a state such as California which has over 13 million housing units [3], this translates to in excess of 65 MegaWatts of

constant extra load, imposing a considerable strain on the state’s power grid. While it is conceivable that future generations of home gateways will implement sleep-on-idle (SoI) capability, this will prove ineffective when the household has devices that generate continuous light traffic. A typical household today is estimated to have between 4 and 7 wireless devices, and this number is estimated to grow to 15 within a few years. Devices for VoIP, continuous health-monitoring, smart-metering, security surveillance, etc. are expected to proliferate that require continuous online presence – this “insomnia in the access” [4] has been shown to severely limit the benefits of SoI in home gateways.

In this work we investigate the feasibility of reducing the combined energy footprint of home gateways by pooling their wireless resources and dynamically aggregating user traffic on to a subset of gateways. Pooling of WiFi resources is already being done by several commercial providers today; Telefonica offers BeWiFi [5] and Comcast offers Xfinity [6]. The density of WiFi access points in urban areas is known to be high (prior work [7] measures between 4 and 6 WiFi networks visible at a typical urban location). It is also known that the average load on an access point is quite low (often $< 2\%$, corroborated by prior studies and our own measurements). This would suggest that in theory it should be feasible to dynamically sleep/wake access points to save energy while still providing connectivity (and adequate bandwidth) to clients. Any such solution however faces several barriers to successful adoption:

- V. Sivaraman and S. T. Ali are with the School of Electrical Engineering, University of New South Wales, Sydney, Australia. Emails: {vijay,taha}@unsw.edu.au
- J. Matthews and C. Russell are with CSIRO ICT Centre, Sydney, Australia. Emails: {john.matthews, craig.russell}@csiro.au
- A. Vishwanath is with IBM Research - Australia. Email:arvishwa@au.ibm.com.

- **Heterogeneity amongst households:** Unlike a centrally managed enterprise WiFi network, a solution that controls autonomous home gateways for coordinated energy savings has to deal with heterogeneity amongst households in WiFi security settings, ISP connectivity, IP address ranges, etc.

- **Diversity of clients:** The solution cannot be specific to client platform or operating system, and should encompass not just today’s platforms (Windows, Linux, OS X, iOS, Android, etc.), but also emerging devices (e.g. IP-TVs, healthcare devices, smart meters, etc.). This requires that no software and/or configuration changes be imposed on clients, lest the barrier-to-entry for participation be too high.

- **User experience:** Dynamic aggregation of clients to WiFi access points (APs) for energy savings will necessitate migration of sessions between APs. The solution therefore has to protect user experience (e.g. minimize video or voice call dropouts) so as to not discourage participation.

- **User fairness:** The overall benefits of lower access network energy footprint, as well as the data costs associated with supporting traffic for non-home clients, have to be shared fairly amongst households.

Although [8] demonstrated a solution for greening enterprise WiFi networks, it does not directly translate to the residential setting as it does not account for factors related to heterogeneity and fairness, as noted above. We are aware of only one existing proposal for greening residential WiFi networks [4] which takes a distributed approach and embeds intelligence in clients to dynamically aggregate their traffic to a reduced set of APs. While the solution is novel, we believe it is impractical for large-scale adoption as it (a) requires complex client-side machinery (including interface virtualization, reverse NAT, traffic snooping, etc.), which imposes a heavy burden on users and does not extend easily to diverse client platforms, (b) falls apart when the network includes non-compliant (careless or malicious) clients, and (c) does not address fairness (in benefits and costs) which is often topmost on participants' minds.

Unlike the approach in [4], in this paper we propose a solution architecture in which centralized control is exercised to achieve network energy savings in a fair manner. The centralized entity could be an ISP or any over-the-top third party. Centralization comes with the benefit of realizing optimal or near-optimal solutions more easily, and fairness can be readily incorporated. Centralization does introduce concerns around failures and attacks. However, we note that there is growing confidence in the research and operations community that these issues are surmountable, and indeed emerging platforms for software defined networking (that use centralized control) are demonstrating that these issues are better handled via software in a centralized way rather than via embedded distributed protocols. Further, centralization obviates the client from implementing new mechanisms, and this transparency removes a major barrier to adoption, since users need only sign-up and forget, leaving the burden of greening the access network to the centralized entity. There is (inevitably) a price to be paid for client transparency – session migrations from one AP to another cannot be seamless and the transient disruptions during hand-off will cause some traffic loss affecting user quality-of-experience. However, as we will demonstrate later, the central controller has visibility into client traffic rates and can therefore choose to perform migrations during periods of “light” traffic, thus minimizing user disruptions. Our specific contributions are:

- 1) We develop a solution architecture for energy savings in today's residential WiFi networks. We argue that our architecture is very suitable for immedi-

ate large-scale deployment since it: (a) centralizes control, enabling for optimality and fairness in energy savings, (b) overlays on today's networks at very low cost, allowing for heterogeneity amongst households, (c) has low barrier to entry, as no user-side management or client modification is required, and (d) enables substantial energy savings with minimal impact on user performance.

- 2) We develop (optimal and heuristic) algorithms that maximize energy savings while enforcing fairness, and demonstrate their performance on real traffic traces taken over 3 days from a building having 30 access points and 25,000 wireless connections, showing how the energy savings vary with algorithm parameters such as fairness weights and client disruption thresholds.
- 3) We implement our algorithm to operate in real-time over a test-bed of 6 commodity residential access points, and test it in a two-storey building emulating 6 households with real VoIP (i.e. Skype), video, and browsing traffic. We quantify the energy savings with power measurements, and show that over 60% of radio energy can be saved with minimal impact on user experience.

The rest of this paper is organized as follows: in §2 we outline our architecture for energy savings under centralized control. In §3 we formulate the optimization problem, develop an efficient algorithm, and discuss its application, while §4 evaluates its performance via simulation of trace data. §5 describes our prototype implementation and results. We discuss related work in §6, and conclude the paper in §7.

2 SYSTEM ARCHITECTURE

A central goal of our solution is to minimize the burden on users to participate in the energy savings scheme. To this end, we argue for a “set-and-forget” approach whereby users opt for the greening service (say by ticking a box during signing up, much the same way utility companies do it today), hand over (some) control of their AP to the green network operator, configure one new SSID on their client devices (explained below), and then use their devices as normal. By not requiring users to install (and maintain) special software on their devices, our scheme is largely transparent to users, encompasses all current and future generation of client WiFi devices, and lowers management, maintenance and support costs for the operator.

The “green operator” (who may be an ISP or a third party, henceforth referred to as “operator”) takes on the responsibility of greening the aggregate of residential WiFi networks. This operator has management control of all the residential gateways (APs), and makes centralized decisions of their states (awake versus asleep), as well as the corresponding client associations. In order to do this the operator installs the following capabilities at the APs (we have done so easily on commodity off-the-shelf APs – e.g. D-Link DIR 615, NetGear WNR2000, Linksys

E2000, TP-LINK WR1043ND) running the DD-WRT [9] open-source firmware:

1) Dual-SSIDs: To delineate “home” clients from “guest” clients, each AP is configured with two SSIDs. The “home” SSID is managed by the user (much like today), and can have name and security settings as desired by the user. Devices that connect to the home SSID are identified by the operator as belonging to that user. The “guest” SSID is common across all APs, and is configured with a common security key (we note that security on the guest SSID is not a big concern since attachment and traffic on this SSID is tightly controlled by the operator). All user clients have to be configured with both SSIDs (with the home SSID being the more preferred network if possible).

2) Client monitoring: The AP has the capability to monitor client activity (traffic flows and their rates) and report these back to the central controller. This information is used by the operator to make the energy savings decisions. Our implementation runs an open source IP traffic logging tool called RFlow (an alternative to the industry standard NetFlow application [10]) at the APs to collect this information. Further, the AP also collects information (by snooping on all channels) about clients that are within range but connected to other APs. This information is used by the operator to deduce feasible alternate paths for each client if migration is necessary. We modified and installed the Wiviz module [11] on our APs to achieve this functionality.

3) Radio management: The operator turns the radio in the home gateway on/off remotely (over the WAN) to save energy. Remote management is commonly available on today’s gateways. In the future, we hope that gateways will have sleep-on-idle (SoI) and remote wake-up capability, which will allow the entire gateway (and not just the radio) to be put to sleep and woken up to enable greater energy savings.

The central controller runs the energy savings algorithm periodically (every 2 minutes in our prototype) to determine the set of APs that need to be on, and the client associations. The algorithm takes as input the set of client connections and their data rates (obtained from RFlow on the APs), the alternate paths available to clients (available from WiViz on the APs), along with running estimates of costs borne by the APs (for fairness). It then computes (details in the next section) the best set of APs to be (kept or turned) on, so as to minimize energy, while limiting client traffic disruptions as well as enforcing fairness (discussed next). The controller then migrates clients between appropriate APs as needed by white-/black-listing their MAC addresses at the corresponding APs (discussed below).

Fairness is needed to ensure that the benefits of energy savings, and the potential data costs of guest client downloads, are shared equally amongst the APs. Centralization makes this relatively easy to achieve. For each AP, the central controller maintains a running cost (in dollars) of supporting “guest” clients (the distinction

between “home” and “guest” clients is easy to make as only the former connect to the home SSID of the AP). This cost comprises two parts: the *energy cost* is attributable to the power consumption of the AP, and is the sum of a base (static) power that the AP consumes when sleeping (i.e. radio is off), and the dynamic power that is the increment when the AP is active (i.e. radio is on), while the *data cost* is associated with downloads performed by guest clients (this information is available from the AP’s RFlow records). Note that home client downloads (i.e. when a client is in its home network) do not count towards this cost, since these are not shared contributions towards the scheme. The guest data cost is balanced across all the users without regard to the volume of each home client download. The combined dollar cost (of energy and data) is averaged for each AP using an exponential averaging technique. This averaged cost is used by our algorithm (detailed in §3) as a selection weight so that an AP that has incurred high guest cost is more likely to be put to sleep to achieve fair distribution of costs.

Migrations (aka hand-offs) of clients are essential, so as to permit the dynamic aggregation of traffic to a reduced number of gateways for energy savings. Unfortunately, unlike in an enterprise environment, migrations entail disruptions in a residential setting, since the client has to obtain a new IP address associated with the AP (and ISP) it migrates to. It is possible to overcome this problem using sophisticated techniques that virtualize the client’s WiFi interface and use reverse NAT to provide a seamless IP address to the application, as demonstrated in [4]. We do not preclude such an approach for clients (and users) that are capable of doing so, but focus on the more likely case of unmodified clients that incur the disruption during migration. The migration is tightly controlled by the central controller by black-listing the client’s MAC address from the AP it needs to be migrated out of, and white-listing it at the AP it needs to migrate to. This allows the controller to track ISP and power costs incurred by guest clients (on the guest SSID). As we will detail in our experimental section, we were able to migrate arbitrary clients (Windows, iPhones, etc.) across APs using this method. To preserve quality-of-experience for the user, the controller can be programmed with the intelligence to effect migrations only during low activity periods, such as when a user has Skype presence but is not actively engaged in a video conference session. Presence applications are designed to be robust to network disruptions (e.g. Skype reconnects within a minute of migration), and this does not therefore pose a major concern.

A concern that we do not directly address (and indeed arises with earlier schemes such as [4] as well), relates to **coverage blackholes**. If turning off a set of AP radios creates a coverage blackhole to which a new client arrives, the client sees no network to connect to. One option is to wake up each AP periodically to allow new clients to join, and in future, wake-on-WirelessLAN (WoWLAN)

technology may solve this problem. But even in the interim, we do not believe this problem is catastrophic. If the household has several devices, some of which demand continuous presence, the controller will ensure that at least the guest SSID is visible to the household. Moreover, since the user has control over the home SSID, they have the option to manually turn on their home AP radio (either with a button or by logging in using a wired port). This “manual override” is an accepted feature in many systems, and so we do not attempt to overcome the blackholes problem algorithmically in our solution.

3 OPTIMIZATION AND ALGORITHM

We now discuss our algorithm for energy savings. We first develop an optimization framework, prove its NP-hardness, and then develop a heuristic algorithm with known bounds. We conclude with a brief discussion on the selection of fairness weights in the algorithm.

3.1 Optimization Framework

Our framework determines the *minimum* set of APs that are needed to provide Internet connectivity to a set of (residential) end-user clients. The APs are numbered 1 to N , and each AP has broadband (i.e. WAN) download bandwidth capacity C_j Mbps for $j = 1, \dots, N$. Let $U = \{1, 2, 3, \dots, n\}$ denote the set of clients, and each client has bandwidth requirement (measured over a certain interval) of b_i Mbps. Denote by e_{ij} the indicator variable that is 1 if and only if client i can connect to AP j on any frequency channel with a specified minimum signal strength (chosen so that the connection is at an acceptable rate, as discussed in §5). Denote by W_j the weight associated with AP j (as briefly described above, these weights are used to control fairness in energy consumption and bandwidth costs across the APs). Our optimization framework takes as input the b_i 's, C_j 's, e_{ij} 's and W_j 's as defined above, and computes $\forall i, j: x_{ij}$ which is 1 if client i is connected to AP j , and 0 otherwise. We also denote by X_j the binary variable indicating if AP j is on or off. The objective function of our optimization is to minimize the weighted power P :

$$P = \sum_j W_j X_j \quad (1)$$

subject to the following constraints:

$$x_{ij} \leq e_{ij} \quad \forall i, j \quad (2)$$

$$\sum_j x_{ij} = 1 \quad \forall i \quad (3)$$

$$\sum_i b_i x_{ij} \leq C_j, \quad \forall j \quad (4)$$

$$\sum_i x_{ij} \leq X_j \sum_i e_{ij} \quad \forall j \quad (5)$$

Constraint (2) ensures that a client i connects to AP j only if the link is feasible. Constraint (3) restricts each client to connect to exactly one AP, while (4) ensures that the sum of the bandwidth requirements of all clients

does not exceed the broadband capacity at the AP. Constraint (5) mandates that an AP is turned on only if at least one client connects to it. We call this problem as minimum weighted set of APs (MWS-AP) that provide network connectivity to all clients.

3.2 NP-Completeness and Heuristic Algorithm

We now formally prove that the decision version of the above MWS-AP problem is NP-Complete:

Theorem 1. *Determining if there exists a solution to the MWS-AP problem above, with weighted power no more than P , is NP-Complete.*

Proof: Given an association of clients to APs (i.e. x_{ij} 's), it is easy to verify that it is feasible (2), connects each client to exactly one AP (3), and does not exceed the bandwidth capacity of an AP (4). APs that connect to at least one client need to be on (5), and their weighted power is verified to be at most P . Therefore MWS-AP is in NP.

To prove that MWS-AP is NP-Hard, we show that the Weighted Set Cover problem can be reduced to MWS-AP in polynomial time. Let the universe of elements be $U = \{1, 2, 3, \dots, n\}$, where each element corresponds to a client. Let $S = \{S_1, S_2, \dots, S_N\}$ denote the collection of subsets of U , where $S_j \subset U$ corresponds to the clients that can connect to AP j . Let W_j be the weight of set S_j , corresponding to the weight of AP j . The above (trivial) transformation is polynomial in time, and it is easy to see that if the MWS-AP (with capacity C_j of each AP set to ∞) can be solved in polynomial time, this will directly yield a solution to the weighted set cover problem. ■

The best known algorithm for weighted set cover has an approximation factor of $O(\ln n)$, where n is the number of elements in the universe. We adapt this greedy algorithm for MWS-AP by selecting at each step the AP that has the highest value of the number of (as yet unconnected) clients divided by the weight of the AP. The algorithm is formally described next.

Algorithm 1 Determine the set of active APs

Inputs: Set of clients U , visible client set S_j and weight W_j for each AP j

Output: Set I of APs that should be active

Temporary variables: X, I

- 1: $X \leftarrow U, I \leftarrow \phi$
 - 2: **Repeat** until $X = \phi$
 - 3: Pick AP j with smallest $|X \cap S_j|/W_j$
 - 4: $I \leftarrow I \cup \{j\}, X \leftarrow X \setminus S_j$
 - 5: **Output** I
-

The algorithm above takes as input the set of clients U , and for each AP j , its coverage of clients S_j and weight W_j . Internal variable X keeps track of the set of clients that are uncovered, and is initialized to the entire set of clients U in step 1. Variable I stores the selected APs, and is initialized to the null set in step

1. The algorithm operates in a loop till all clients are covered, i.e. $X = \phi$ (step 2). In each iteration, the AP j which has the maximum ratio of unconnected clients ($|X \cap S_j|$) to weight (W_j) is selected (step 3). This AP is added to the set I of selected APs and the clients S_j it covers are removed from X (step 4). The set of active APs I is output in step 5.

3.3 Client Migrations and AP Fairness

Our algorithm determines the set of APs that need to be on, but does not clarify the attachment of clients to the APs. We therefore refine our algorithm in the following ways:

- For each “active” client, i.e. one whose traffic intensity is above a certain threshold (we will study the impact of this threshold in the next section), we do not want to disrupt its existing connectivity as it can impact user quality-of-experience. Our algorithm therefore begins by keeping such APs on, i.e. in step 1 it will initialize the set I to include such APs, and the set X to exclude clients covered by these APs.
- The algorithm outputs the set of APs that need to be on. A client may have multiple APs within range that are on, and we use the following method to decide which one it connects to: (a) if the client’s home AP comes on, it is migrated to its home SSID (as it is unnecessary to incur a guest cost when the home is available), (b) otherwise, if the client’s current AP is on, the client stays connected to the same AP and is not migrated (so as to minimize migration disruptions), and (c) if neither the home AP nor the current AP of the client are on, the client is migrated to the AP with least weight (for fairness, as described next).

The determination of fairness weights was discussed in §2, here we formally describe how it is computed. The AP weights W_j are updated each time the algorithm is executed, every $T = 120$ seconds in our studies. Denote by P_b the base power consumption of the AP (with radio off, the TP-LINK AP used in our experimentation consumes $P_b = 2.8$ W), and by P_r the additional power consumed when it is on (the radio on the TP-LINK consumes $P_r = 1.7$ W). Then the *base cost* of AP j over the interval of T seconds is $C_p P_b$ where C_p is the unit cost of energy (we used the energy rate of \$0.23 per KWh). The *running cost* of operating AP j over the interval comprises (a) the energy cost of $C_p X_j P_r$ where X_j is the indicator variable denoting if the AP (radio) was on or not over the interval, and (b) the data cost $C_d D_g$ where C_d is the unit cost of data (we used \$1 per GB), and D_g denotes the data download in KB by guest clients attached to this AP over the interval. We use a linear cost function for data cost to exemplify how it can be introduced in our scheme. The cost function can be suitably adapted depending on the deployment environment. The weight of AP j is then updated by combining the base cost and running cost as:

$$W_j^{update} = C_p P_b + \beta(C_p X_j P_r + C_d D_g) \quad (6)$$

The scaling parameter β can be adjusted by an operator to control fairness. If the operator sets β to zero, then running costs become irrelevant, and overall energy savings is maximized without being fair. If, on the other hand, the operator wants to implement (and market) fairness for their users, they can easily do so by appropriately choosing this parameter. If $\beta \rightarrow \infty$, running costs will be more fairly shared, at the cost of reducing global energy savings. This trade-off between fairness and efficiency will be investigated in the next section. Lastly, we note that the weight W_j is averaged over time t using:

$$W_j[t] \leftarrow (1 - \alpha)W_j[t - 1] + \alpha W_j^{update} \quad (7)$$

The exponential averaging technique above is routinely used as a way of obtaining longer term averages without explicitly maintaining window size of samples. The parameter α controls the time period over which the averaging is done – for our study that uses traces which are 24-hours long, we wanted to enforce fairness over a window of a few hours, and so we chose $\alpha = 0.01$, which reduces the importance of a new sample (computed every 2 minutes) to half its value in about 2 hours. If fairness is to be enforced over longer time periods (e.g. many days), a lower value for α can be used.

4 SIMULATION USING TRACE DATA

We now study the efficacy of our algorithm by applying it to real trace data. Though our algorithm is designed for home networks, obtaining WiFi data from sufficient households to test the algorithm at scale is very challenging (the next section will describe our prototype experimentation over a small test-bed of 6 WiFi access points). To validate our algorithm at larger scale, we use traces taken from a university campus WiFi network. We first describe the characteristics of the campus WiFi network and the data traces, and then describe the energy-savings trade-offs obtained from simulating our algorithm over these traces.

4.1 WiFi Network and the Trace Data

The university campus comprises over 850 APs, serving tens of thousands of client connections on a typical day. To make our study tractable, we chose to focus on just one building. This building has 6 levels (Lower Ground, Ground, and Levels 1-4), and can be thought to be representative of a multi-storey dwelling unit comprising many households. The building houses 30 APs, which are Cisco Aironet 1140 Series Dual-Band Access Points (model AIR-LAP1142N-N-K9) with a power draw of approximately 15.4 W each (12.95 W base power, with an additional power draw of up to 2.45 W for Power-over-Ethernet depending on the length of the interconnecting cable). Thus the APs in this building consume about 462 W of continuous power, and we seek to quantify the potential for power savings on a typical day, along with the impact on user experience.

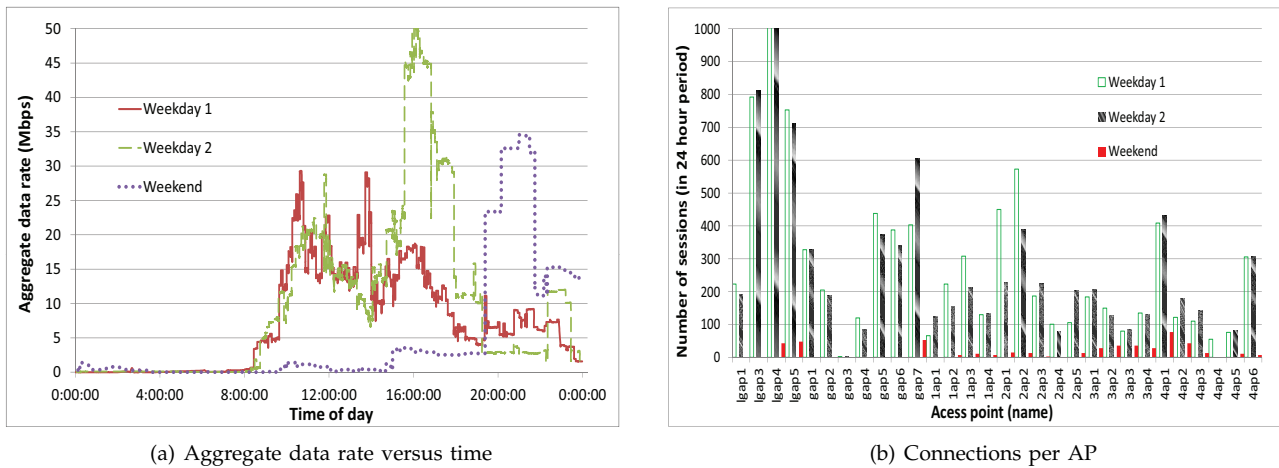


Fig. 1. WiFi statistics for 2 weekdays and 1 weekend: (a) aggregate data rate of clients over a 24-hour period, (b) number of client connections for each of the 30 APs over a 24-hour period.

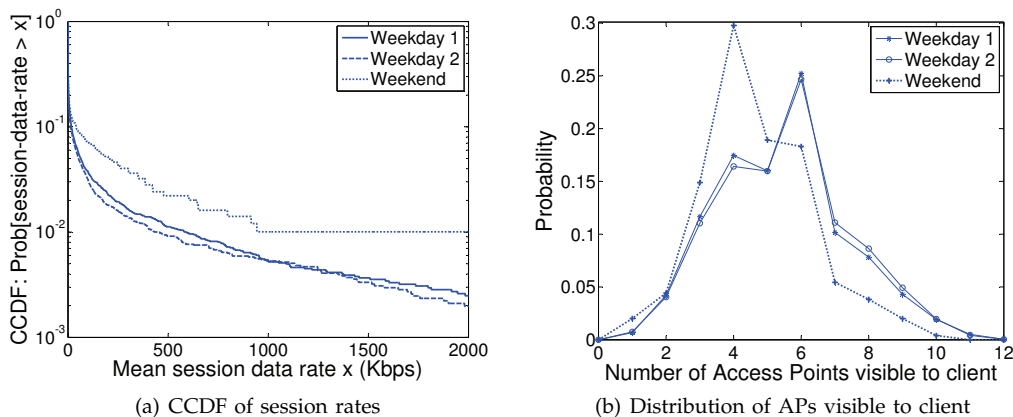


Fig. 2. (a) CCDF of mean client session rates for 2 weekdays and 1 weekend, and (b) Distribution of number of APs visible to a client for 2 weekdays and 1 weekend.

For the purposes of this study, we obtained root access to the wireless control system, which allowed us to capture detailed statistics on all wireless client connections in the building over several consecutive days. Specifically, we analyzed traces spanning a week in April 2012. Each weekday had approximately 8500 client sessions (across all 30 APs in the building), while the weekend had around 500 sessions. The traces showed that the demand for WiFi in the building varies significantly depending on the time of day, with 300-400 clients concurrently connected during peak hours on a typical weekday, while only a small handful are connected during off-peak hours. Fig. 1(a) shows the aggregate data rate of the clients over a 24-hour period (for 2 weekdays and 1 weekend), typically in the few tens of Mbps during peak hours on weekdays, and no higher than 1 Mbps during off-peak hours. The distribution of load across the APs is non-uniform. Fig. 1(b) shows as a bar graph the number of client sessions (connections) at each of the 30 APs over the 24-hour period. Some of the APs in the lower ground floor (prefixed “eeblg”) see over 800 connections on a weekday (as it houses the student computer labs) while some of the APs in levels 3 and 4 see less than 100 sessions.

We also have access to the mean data rate in each client session. This is plotted in Fig. 2(a) as a complementary cumulative distribution function (CCDF), and shows a rapid drop-off in the initial part. Nearly 70% of the sessions had an average data rate below 1 Kbps, and over 90% of sessions have average data rate below 20 Kbps. This confirms earlier observations that several clients generate light continuous traffic (such as to maintain presence), and can easily be aggregated onto a smaller number of APs to save energy. However, the ability to aggregate traffic requires clients to be able to connect to alternative APs. To determine this, we measured the coverage area of each AP in the building (corroborating it against the coverage maps provided by IT services), and determined the overlap in coverage for every pair of APs. Using these, we deduced the probability with which a client can connect to other APs in the building. Note that the coverage area is purely required for simulation purposes. A real system does not need to know the coverages. As we will show in the next section, the APs discover the connectivity dynamically using the WiViz module. The probability distribution of the number of APs to which a client can connect to is plotted in Fig. 2(b) for the client traces of each of the three days, and shows

that over half the clients can see 6 or more APs. The average number of APs visible to a client was found to be 5.8, consistent with the number observed in residential networks by earlier studies [7].

4.2 Results and Discussion

Using the above WiFi traces from our building containing 30 APs, and spanning 25,000 client connections over three days, we determine the potential for energy savings and the associated trade-offs in terms of session disruptions and fairness in costs.

4.2.1 Energy Savings and Client Migrations

As described in §2, our scheme requires no client software changes, but entails transient disruptions to client sessions as they are migrated between APs for energy savings. These can cause annoyance for users if they are active on their device, but are not problematic for presence applications (e.g. Skype) which automatically reconnect after network disruptions. To understand the trade-off between energy savings and migrations, we simulate our algorithm over the traffic traces for different values of threshold θ , which is the value of the session rate (in Kbps) above which the client is not allowed to be migrated (i.e. the client is deemed “active”). Fig. 3(a) shows the energy savings as a function of θ for each of the three days, while Fig. 3(b) shows the average number of migrations incurred by a client session. We make several observations from these figures: (a) When the threshold is set to $\theta = 0$, the algorithm is essentially disabled, since no migrations are permitted – this allows APs to sleep only if they have no clients connected, i.e. corresponds to sleep-on-idle (SoI) capability. The energy savings with SoI are approximately 48% on weekdays and 78% on weekends. Though this may seem encouraging, it arises because of the uneven distribution of clients to APs in this building, and we believe the SoI savings are likely to be much lower in the residential setting wherein most homes in the near future will have always-on devices such as sensors, healthcare monitors, etc. (b) As the threshold θ increases, so do the energy savings, since clients can be migrated more freely to aggregate them at a smaller number of APs to allow greater energy savings. When no barrier is placed on client migration (i.e. $\theta = \infty$), energy savings increase to over 70% on weekdays and 87% on weekends, which is substantially higher than the SoI energy savings. However, this comes at the cost of increasing the number of migrations per client session, rising to 1.2 on weekdays and 1.4 on weekends. The curve of energy savings seems to have a knee around the 10-20 Kbps mark, suggesting that threshold θ in the range 10 to 20 Kbps allows much of the energy gains to be obtained, while the number of migrations per-session is relatively low. We therefore use θ in this range for our subsequent study.

4.2.2 Fairness

Prior studies have not considered the fairness of cost-sharing among APs. Fairness is a significant issue in

users’ minds, so they know that everybody in the community is contributing equally. The AP weights W_j in our algorithm are used to provide fairness, and the parameter β in (6) can be increased to make the system more fair, as described in §3.3 above.

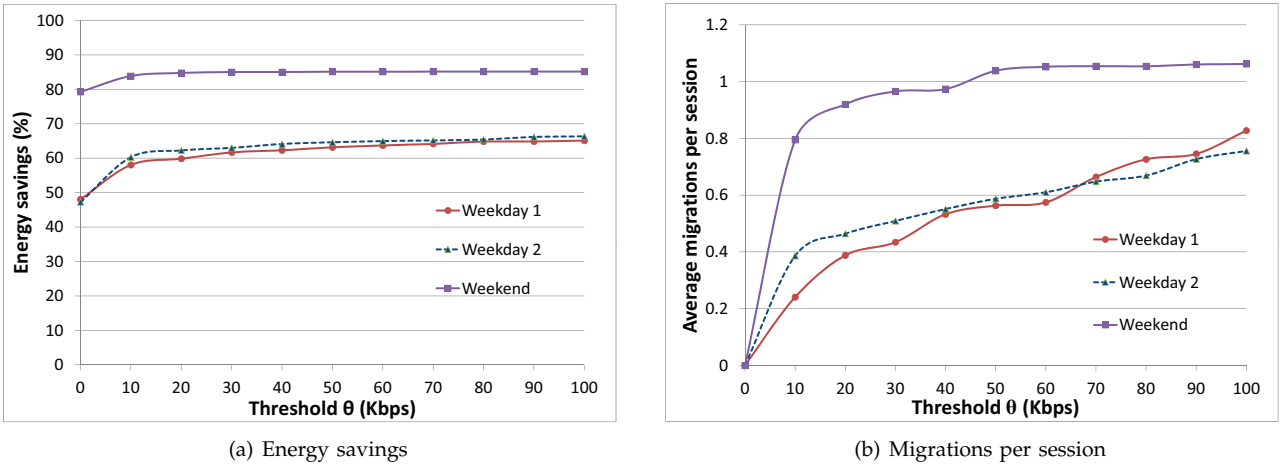
In Fig. 4 we depict the impact of parameter β on energy savings (left axis) and unfairness (measured in terms of the standard-deviation of the average guest traffic rate) across APs (right axis) for a weekday (Fig. 4(a)) and a weekend (Fig. 4(b)). For the setting $\beta = 0$, running costs of APs are ignored, causing all APs to have the same weight in the algorithm. This yields best savings in overall network energy, but is not as fair – for example, the four APs on the lower ground level respectively supported 3.5, 5.4, 1.5, and 0.2 Kbps of average guest client traffic during the day, implying that they incur very different costs for participating in the algorithm. This unfairness can be quantified in terms of the spread (i.e. standard deviation) of the guest traffic rate supported by the APs, and for $\beta = 0$ (no fairness), the plot shows this to be over 1.5 Kbps on a weekday and 0.65 Kbps on a weekend. When β is increased (towards 1), the running cost of supporting guest traffic at each AP plays a dominant role in determining its weight, which leads to increased fairness. The same four access points on the same day would now support 1.6, 1.6, 1.4 and 0.7 Kbps of average guest client traffic, and the costs are more evenly spread across all APs as shown by the bottom curves in Fig. 4. The increased fairness with parameter $\beta = 1$ reduces the standard deviation in guest client rates supported by the APs to nearly 0.8 Kbps for the weekday and 0.5 Kbps for the weekend. Note that the layout of APs and clients may not allow for perfect fairness (we increased β to large values and found that the spread did not reduce much beyond what we got for $\beta = 1$); for example, an area of the building with higher density of APs will allow for higher fairness (since the APs can take turns to cover the area) whereas an area covered by just one AP prevents it from ever being turned off, causing unfairness. In general, denser deployments permit greater fairness, and this is corroborated in our experimental evaluation in the next section. We also note that increased fairness comes at the expense of reducing the overall energy savings (blue curves in Fig. 4), though for this network the reduction is marginal.

5 SYSTEM EVALUATION

We prototyped our system on commodity hardware and deployed an experimental test-bed in a 2-storeyed building representative of 6 apartments in a multi-dwelling complex. The hardware/software configurations, test-bed setup and performance results are described next.

5.1 Hardware and Software Configurations

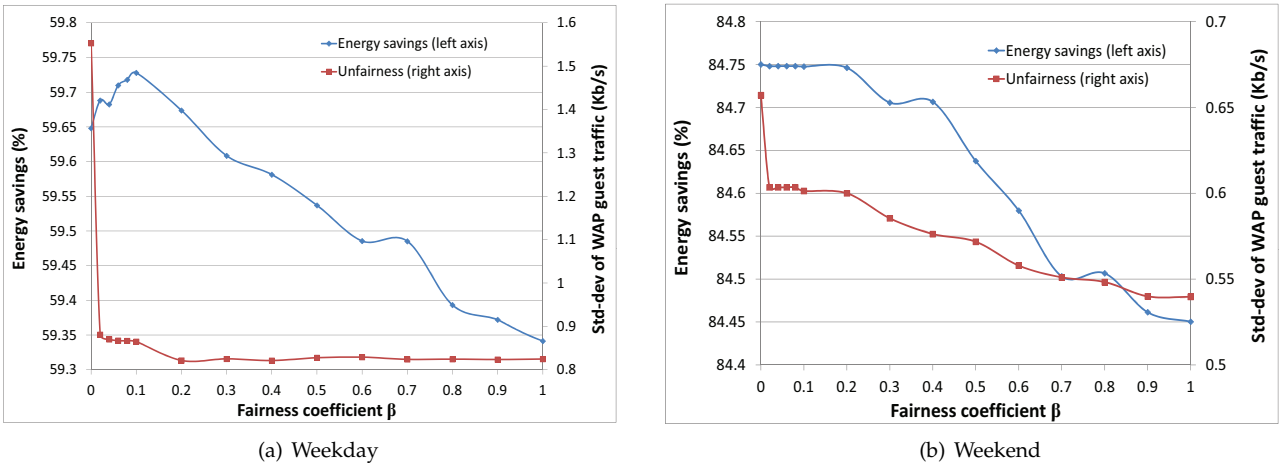
Access Points: We measured the power consumption of several popular residential APs, and found that the radio



(a) Energy savings

(b) Migrations per session

Fig. 3. Algorithm performance for campus WiFi traces as a function of migration threshold θ : (a) Energy savings for 2 weekdays and 1 weekend, and (b) Average migrations per session for 2 weekdays and 1 weekend.



(a) Weekday

(b) Weekend

Fig. 4. Overall energy savings (left axis) and unfairness (standard deviation of guest traffic, right axis) as a function of parameter β for (a) Weekday and (b) Weekend.

accounted for between 20-40% of their overall power consumption (note that our current prototype turns only the AP radios on/off, future hardware that supports wake-on-WAN capability can allow the entire AP to be put to sleep and woken up remotely). For our test-bed we used the TP-LINK WR1043ND [12] AP, which consumes 4.5 W with the radio on (this was largely invariant to traffic load) and 2.8 W with the radio off – the radio thus accounts for nearly 40% of its energy. We installed the DD-WRT Linux-based open source firmware [9] on it and enabled data collection capability with RFlow. Further, the set of feasible migrations for each client was determined with the help of the Wiviz module [11].

Database: We configured each AP to export NetFlow records of wireless clients to a remote database at 1 minute intervals, thereby logging all IP traffic flows for these clients passing through all the APs. These records include information such as timestamps, source and destination IP address, IP protocol, source port, amount of data in each flow, etc. The APs also send updates to the controller indicating the identities of APs and clients

within radio range. We store all this information in a MySQL database at the controller and use it as input to the algorithm. We used Wireshark to quantify the amount of communication between the controller and the APs, and noted that the data rate between each AP and the controller varied between 0.18 and 1.5 Kbps, implying that the benefits of centralization do not come at expense of large communication overheads.

Controller: The controller functionality was programmed in Java, running on a FreeBSD machine. At 2 minute intervals, the controller checks the flow records for each client to determine those that are *Idle* (i.e. have traffic rate below the Idle threshold), and hence suitable for migration. It also checks the records for each AP to determine the clients that can be covered by that AP, as well as traffic and power consumption of each AP for fairness calculations. It then runs the algorithm to determine the most suitable set of APs (radio) to keep on, and the most appropriate AP to serve each client. The controller then turns each AP radio on/off by logging in to the AP over its WAN port via an SSH script.

5.2 Experimental Setup

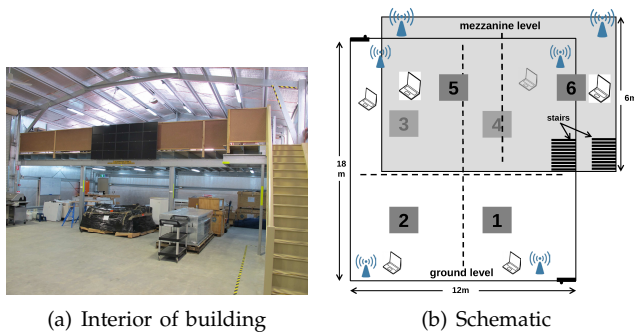


Fig. 5. Experimental setup: (a) Two-level 12m x 18m building emulating six residential apartments, (b) Schematic (top view) of the building.

Test site: We deployed our experimental test-bed in a two-level 12m x 18m shed (pictured in Fig. 5(a)) that allows emulation of six residential apartments with four units in the lower floor and two in the upper (schematic shown in Fig. 5(b)). Each apartment is outfitted with one AP. The APs draw power from a common source, and we used a power meter to measure the aggregate power consumption of all APs. The power is logged every second, and exported to a PC for later analysis. The central controller runs off-site.

AP and client setup: As mentioned in §2, APs and client devices are configured with two SSIDs. Further, each AP has an Ethernet (i.e. WAN) broadband link and runs its own DHCP server. Our prototype allows the APs to operate on arbitrary channels, since the Wiviz module is able to scan for clients on all frequency bands. Our implementation does not currently consider the rate at which a client connects to an AP. It has been found that a client with a weak signal strength connecting to an AP can reduce the rate for all other clients connected to that AP, resulting in a rate anomaly [13], [14]. Though our implementation does not currently handle the rate anomaly, the centralized nature of our algorithm allows this to be relatively easily incorporated as a constraint in the optimization, by restricting connectivity to clients that have a sufficiently high signal strength (obtained from WiViz information). Lastly, our algorithm can be further refined to include frequency planning and dynamic adaptation of AP radio signal strength; however these enhancements are left for future work.

Our client devices are MacBook Pro machines running Windows 7, one per household. On each client the home SSID is configured as a higher priority network than the guest SSID, to ensure that the client chooses to connect to the former when both are available (this helps speed up the migration of a client to its home SSID, but is not strictly necessary since our MAC filtering will prevent it anyway). We *emphasize* that no special software or setting (other than the home and guest SSID configuration) is needed on the clients, making our scheme amenable for any breed of client device (PCs, smart-phones, medical

devices, etc.) running any operating system (Windows, OS X, iOS, Android, etc.).

Client migrations: The controller periodically executes the algorithm to decide which AP radios should be turned on/off. It can migrate clients from one AP to another when turning an AP off, or for fairness in load across APs. In our network, client migrations are achieved by black-listing the client’s MAC addresses at its current AP and white-listing it at the AP we want it to connect to. This is only done for the guest SSID and does not hinder the addition of new clients to their home SSID. When a client migrates between two separate SSIDs, such as from home to guest or vice versa, the client TCP/IP stack automatically obtains a new IP address by sending a DHCP RELEASE followed by a DHCP DISCOVER. When migrating a client between APs on the same (guest) SSID, we found that the (Windows) client attempts to keep its IP address (as typically happens in an enterprise WiFi network) by first sending a broadcast DHCP REQUEST; only when this times out after 5 attempts does the client send a DHCP DISCOVER to obtain a new IP address from the AP it has migrated to. In either case, the migration time from one AP to another was found to be between 10 and 20 seconds over 50 measurements. The migration time was not dependent on the number of available APs. The variation in migration times is due to the Windows software and the randomized timers associated with the various DHCP messages that need to be exchanged.

User Traffic: We wrote scripts for each client to generate traffic representative of two types of users – (a) heavy users (who consume over 80 GB of data per month) [15], and (b) typical users (who consume on average about 5 GB of data per month) [16], [17]. In both cases, we model the user as being in one of four states: in *Browsing* mode, the user downloads several web-pages on a browser (at an average bit-rate of 478 Kbps); in *Video* mode, the users play videos from YouTube (average bit-rate 1.528 Mbps); in *Skype* mode, the user initiates a Skype voice session with a random client in our network (average bit-rate 64 Kbps); and finally in *Idle* mode, the client is inactive but maintains online Skype presence (average bit-rate 1 Kbps). These actions are modeled as distinct states in a Markovian process, where the user state transitions in every time step depend on pre-specified transition probabilities. To emulate realistic user behavior, we divide the day into distinct epochs. For the scenario corresponding to heavy users, the day is divided into four epochs – in the morning (6 AM to 9 AM) there is moderate traffic from a client as the user browses and places some voice calls; during the day (9 AM to 6 PM) usage is relatively low; between 6 PM and 12 AM is peak time when users actively engage in voice, video, and browsing; and finally at night (12 AM to 6 AM) Internet use is very low (maintaining presence). In the case of a typical user setting, we divide the day into two epochs – between 8 AM and 2 AM users are either active (i.e. engaging in voice, video and browsing

sessions), or inactive (i.e. in the idle state), and between 2 AM and 8 AM users are always idle and therefore Internet activity is very low (only maintaining presence). We chose the state transition probabilities (for both types of users) to obtain the average fraction of time spent in each state during each time epoch as per Table 1 and 2.

Epoch	Skype	Video	Browsing	Idle
6am-9am	28%	20 %	36 %	16 %
9am-6pm	14%	14 %	27 %	45 %
6pm-12am	29%	32 %	34 %	5 %
12am-6am	1%	2 %	2 %	95 %

TABLE 1

Percentage of time spent in a state by heavy users

Epoch	Skype	Video	Browsing	Idle
8am-2am	1.8%	1.5 %	5.7 %	91 %
2am-8am	0%	0 %	0 %	100 %

TABLE 2

Percentage of time spent in a state by typical users

Metrics: We ran the experiment over 6 hours (of real time), emulating a 24 hour cycle. The central controller runs our algorithm every 2 minutes and accordingly migrates clients and turns APs on/off. Based on results from our simulation study above, we tried several migration thresholds θ in the range 10-20 Kbps – thus any host with a data rate above this threshold is deemed “active” and not migrated, while clients with data rate below this threshold are “inactive” and eligible to be migrated to the guest SSID. We measure the power consumption (using the power meter) every second during the experiment, and also the traffic flow at each AP (i.e. from the AP to each of the clients). For enforcement of fairness, the controller tracks the *guest cost* per AP, which comprises the energy cost incurred by an AP for supporting guest clients, and data costs incurred by the AP for downloads by guest clients. We assume a data cost of \$1 per GB (for ADSL [18]) and a power cost of \$0.23 per kWh. The controller attempts to enforce fairness in guest cost incurred by the various APs (households) by manipulating the AP weights in the algorithm as described earlier.

5.3 Experimental Results

We ran our algorithm over several days. Each run lasted 6 hours, with the traffic pattern emulating a 24-hour cycle as described above. We quantify the performance of our prototype system in terms of power savings and disruptions (§5.3.1), as well as fairness (§5.3.2).

5.3.1 Power Savings and Disruptions

Fig. 6 depicts the average aggregate power consumption of all the APs (in a chosen experimental run spanning 6-hours, emulating a 24-hour cycle) measured using the power meter. Without our power-saving algorithm, the 6 APs together (in the top plot corresponding to heavy users) consume roughly 28.7 W continuously, shown

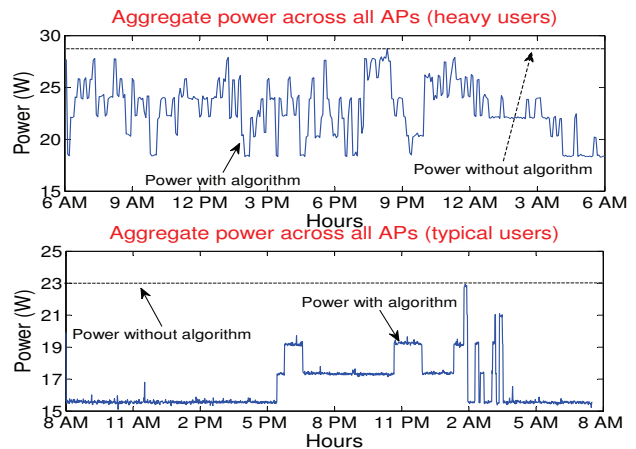


Fig. 6. Average aggregate power consumption of the APs for heavy users (top) and typical users (bottom).

by the dashed line in the plot. By contrast, our algorithm dynamically aggregates clients to a reduced set of APs, and the average aggregate power consumption is reduced to 22.9 W. This is a saving of about 0.97 W per AP, or about 57% of the power required by the AP radio, which is substantial. The power savings become more significant when considering typical usage patterns (the bottom plot). The 5 APs (the power supply of one of the previously used APs died, and therefore became unusable) consume a constant 23 W without the algorithm, and this reduces to 16.5 W with our algorithm running, representing a saving of about 1.3 W per AP, or about 77% of the power required by the AP radio. Over several runs we found the standard deviation of power savings to be between 1.42 W and 3.3 W. Note that (unlike the simulation study earlier) sleep-on-idle (SoI) gives no energy savings, since the clients maintain continuous Skype presence (even in the Idle state). Needless to say, the overall energy savings would be even higher if the AP had remote wake-up capability allowing the entire AP to be put to sleep.

Action	Disruptions (heavy users)	Disruptions (typical users)
Skype Calls	64/284 = 22.5%	14/182 = 7.7%
Video Views	24/257 = 9.3%	2/139 = 1.4%
Browsing Sessions	28/966 = 2.9%	2/551 = 0.4%

TABLE 3

Effect on user experience – heavy and typical users

The energy savings come at the cost of disruptions to user sessions arising from migrations, as depicted in Table 3. We find that in the case of heavy users (column two) more than 22% of Skype calls experience a dropout. Although the disruption is temporary and the user can redial, it does cause some annoyance. We investigated why this was happening and found the following: Skype chooses from a large number of codecs and can have a highly variable bit-rate, ranging from 11 to 230 Kbps [19]. Consequently, there is no one bit-rate threshold at which the controller can reliably detect an “active” VoIP

call, and hence can falsely believe the client to be idle even when it is on the call; this leads to migrations during an active call, resulting in dropouts. As future work we intend to improve algorithm intelligence for detecting VoIP calls to minimize such disruptions. The table also shows that few (9.3%) video views experience some stalls, and a small percentage (2.9%) of the browser downloads timeout due to the disruption. These happen because the controller makes the decisions based on client data-rates that are obtained (using the flow-tools [20] NetFlow collector) only once a minute – if the video/browsing session starts towards the latter part of that minute, the average client rate over that minute is still low enough that the controller falsely deems the client “inactive” and eligible for migration.

The disruptions incurred by Skype, video and browsing sessions are substantially lower in the case of a more typical residential user (column three), implying that the nature of user activity does impact quality-of-experience. 7.7% of Skype calls dropout, while only a very small fraction of video (1.4%) and browsing sessions (0.4%) are affected. In future enhancements, disruptions can be greatly minimized for improved user quality-of-experience by polling data rates more frequently to rapidly detect client activity at the flow level (VoIP, video, etc.).

5.3.2 Fairness

We have already emphasized that it is important to have fair sharing of the benefits of the scheme. Our implementation keeps track of the “guest cost” borne by each AP, comprising the energy cost and the data cost, and incorporates this into the algorithm weights (as described in §3.3) to balance the contributions by each AP. In Fig. 7 we show (left plot) how the cumulative guest cost for each AP evolves over the 24-hour duration in a run for heavy users. It is seen that the costs stay reasonably balanced for the most part, except for sudden spikes that can create transient imbalance. This can be explained as follows: when an inactive client suddenly becomes active, it can impose a significant data download cost on a guest AP (hence the spike). This spurt of activity takes a couple of minutes to be detected by the controller (since it obtains flow records only at periodic intervals), upon which it attempts to migrate the client back to its home network. We notice in the plot that the period 6pm to midnight has several such spikes (being peak hour), which make the costs imbalanced temporarily, but will average out over longer periods (several days). Another “view” of how fair our algorithm is can be seen from the right plot in Fig. 7 which shows the histogram of the aggregate energy and data costs (comprising guest and home clients) for typical users over the 24-hour duration. It can be observed that the power and guest data costs incurred by each AP are well balanced across all 5 APs, demonstrating that the algorithm achieves good fairness even though home usage data costs are significantly different among the

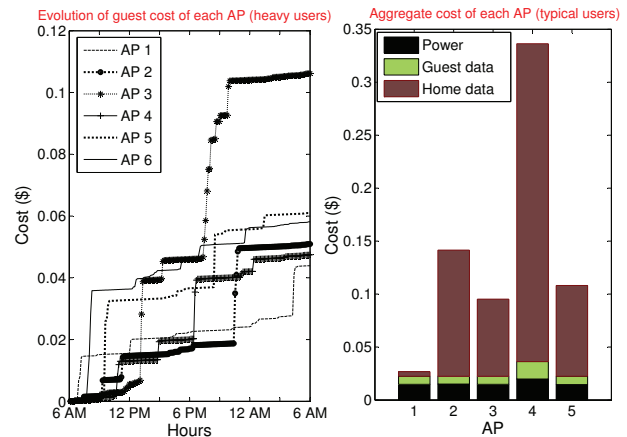


Fig. 7. Fairness in guest cost, i.e. sum of the energy cost and the data cost of the APs for heavy (left) and typical (right) users.

users. The centralized nature of the controller leaves open the door to more sophisticated algorithms that can learn user activity patterns and predictively migrate clients to their home network ahead of activity spurts, thus eliminating spikes and achieving perfect fairness.

6 RELATED WORK

We now provide a brief overview of recent work in the context of reducing energy consumption of access networks, centralized control and AP-level coordination of WiFi networks.

Wireline access networks: Bianco et al. [21] present trends from a telco-perspective (Telecom Italia) and discuss the importance of minimizing energy consumption for enabling sustainable next generation FTTx access networks. Greening DSL networks has also received widespread attention – Tsiaflakis et al. [22] and Guenach et al. [23] revisit the dynamic spectrum management problem in DSL systems and make it power-aware by incorporating constraints for limiting the transmit power. Their results indicate that there exists a tradeoff between power savings and data rates. The problem of (re)designing DSL networks to minimize power consumption is studied by Bhaumik et al. [24] who show that the energy efficiency of access networks can be improved by replacing large monolithic DSLAMs with small DSLAM units closer to the customer.

Mobile devices: Based on the observation that WiFi radios in smartphones consume significant amount of power when active, Rozner et al. [25] propose NAPman, a system to minimize WiFi energy consumption in mobile devices. Although PSM (Power Save Mode) is part of the WiFi standard, the authors note that competing background traffic can adversely impact energy consumption. Consequently, an energy-aware fair scheduling algorithm is developed yielding up to 70% savings. Manweiler et al. [26] argue that NAPman is most effective when a single access point is connected to multiple clients. Since in practice a client is within the range of multiple access points, this scenario strongly impacts energy consumption. A new system called Sleepwell is

designed and savings of up to 51% is demonstrated. Finally, Zhang et al. [27] found that over 60% of the WiFi energy is consumed when the device is in the idle listening (IL) state, even in the presence of PSM. While prior work minimizes the duration a client spends in IL, their proposal E-MiLi reduces the power consumption of IL by reducing the clock rate of the radio when the device is in IL. Results show savings of $\approx 44\%$.

Wireless access networks: Jardosh et al. [8] and Goma [4] et al. demonstrate greening enterprise and residential WiFi networks respectively. However as argued in §1, the former does not apply to the residential setting because it does not address heterogeneity and fairness issues, while the latter uses a distributed approach and places the onus on end-users, which is fundamentally different from our technique that is centralized and controlled by an operator.

Aggregating bandwidth: As the energy saving schemes primarily rely on bandwidth from neighboring wireless gateways being pooled and shared across many users, we now briefly mention few works that achieve this aggregation. Jakubczak et al. [7] and Tan et al. [28] propose Link-alike and CUBS respectively to effectively aggregate and share the uplink bandwidth of neighboring gateways. Their results indicate that the throughput of upload intensive applications improve by more than 30%. Giustiniano et al. [29] show that fairness is an important metric that needs to be factored in when aggregating capacity from multiple access points, which was not considered by Kandula et al. [30]. As a solution, they propose a system called THEMIS that incorporates fairness and addresses the problem in a distributed manner without requiring any changes to the network. COMBINE by Ananthanarayanan et al. [31] is a system that significantly improves the download speed at any node by pooling together bandwidth from the wide area network. OpenRadio [32], based on the software defined networking paradigm, advocates pooling together resources from various wireless access technologies, i.e. WiFi, 3G, LTE etc., for improved network performance and ease of management. It is important to note that while the above works aim to maximize system throughput, reducing the energy consumption has not been their primary focus.

Centralized control and AP coordination: Centralized control has been efficiently leveraged in enterprise WLANs for fault management [33], power control and channel assignment of APs [34], and client localization [35]. Shrivastava et al. [36] develop a framework called CENTAUR for mitigating hidden or exposed terminal interference in enterprise WLANs. CENTAUR is shown to improve the performance of the network substantially due to improved utilization of the wireless medium. Much like our scheme, the centralization functionality of CENTAUR is implemented in a single central server and requires minor modifications to the APs. No changes are needed to the end-user client devices. A system called FLUID was built by Rayanchu et al. [37]

that also employs a central controller and assigns the centre frequencies and channel widths to the APs dynamically, based on traffic demand. Flexible channelization combined with data scheduling is shown to further improve performance. Experimental results demonstrate that FLUID improves the median throughput by 59%. Akella et al. [38] develop an algorithm called Power-controlled Estimated Rate Feedback (PERF) for improving end-user performance in chaotic (i.e. unplanned and unmanaged) dense WiFi network deployments. Home WiFi networks belong to the category of chaotic networks. The key idea behind PERF lies in automatic management of transmission power levels and rates of APs and clients so that interference between neighbouring APs is minimized. Experimental results highlight the benefits of PERF; a client of a highly utilized AP located next to another such pair sees a 20-fold increase in throughput. In essence, this work shows that it is possible to improve the performance of WiFi networks if the APs can be made to coordinate with each other. In the context of home networks, Patro et al. [39] deploy an infrastructure called WiSe to measure and monitor the performance of home WiFi networks. The APs are configured with specialized software and communicate with WiSe's measurement controller using open APIs. Analysis of traces from 30 households spanning over 6 months show that a majority of the links performed well but poor quality was observed about 2% of the time. Finally, Manweiler et al. [40] develop a tool called RxIP for monitoring the health of residential WiFi networks. Specifically, the APs announce their IP addresses periodically. Neighbouring APs that receive these messages then relay them so that they are received by potential hidden terminals. As new APs are discovered, neighbouring APs establish a control channel with these APs over the wired Internet, thus enabling coordination between APs. Results show an improvement of 57% in the median throughput in symmetric hidden terminals.

7 CONCLUSIONS

Residential broadband access gateways are a major contributor to overall network energy consumption due to their widespread deployment. In this paper, we proposed, evaluated and prototyped a scheme for aggregating users on to a fewer set of WiFi access points to reduce energy consumption. We made the following contributions: (1) We developed a centralized architecture that works across heterogeneous ISPs and clients, and allows for fairness in energy savings, (2) We developed (optimal and heuristic) algorithms, and also studied the tradeoff between energy savings and session disruptions using campus WiFi traces (comprising of 30 APs and 25,000 connections), and (3) We prototyped our algorithm in an experimental test-bed consisting of 6 APs and real applications, and showed energy reductions of over 60% with limited impact on user experience. Our scheme is centralized putting the onus on the operator rather than the user, thereby significantly reducing the barrier-to-adoption for wide-scale deployment.

REFERENCES

- [1] R. Tucker, J. Baliga, R. Ayre, K. Hinton, and W. Sorin. Energy Consumption of IP Networks. In *Proc. European Conf. on Optical Communications*, Belgium, Sep 2008.
- [2] M. Pickavet. Energy in ICT – Trends and Research Directions. In *IEEE Advanced Networking and Telecommunications Systems (ANTS)*, India, Dec 2009.
- [3] Number of housing units in California in 2011. <http://quickfacts.census.gov/qfd/states/06000.html>.
- [4] E. Goma et al. Insomnia in the Access or How to Curb Access Network Related Energy Consumption. In *Proc. ACM SIGCOMM*, Canada, Aug 2011.
- [5] BeWiFi from Telefonica. <http://www.bewifi.es/>.
- [6] Xfinity from Comcast. <http://www.comcast.com/>.
- [7] S. Jakubczak et al. Link-alike: Using Wireless to Share Network Resources in a Neighborhood. *ACM SIGMOBILE MC2R*, 12(4):798–814, Oct 2008.
- [8] A.P. Jardosh et al. Green WLANs: On-Demand WLAN Infrastructures. *Springer Mobile Networks and Applications (MONET)*, 14(6):798–814, Dec 2009.
- [9] DD-WRT. Open source linux-based firmware. www.dd-wrt.com.
- [10] Cisco IOS NetFlow. www.cisco.com/go/netflow.
- [11] Wiviz. www.dd-wrt.com/wiki/index.php/Wiviz.
- [12] TP-LINK WAP. www.tp-link.com/en/products/details/?model=TL-WR1043ND.
- [13] R. Murty, J. Padhye, R Chandra, A. Wolman, and B. Zill. Designing High Performance Enterprise Wi-Fi Networks. In *Proc. USENIX NSDI*, USA, Apr 2008.
- [14] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proc. IEEE INFOCOM*, USA, Mar/Apr 2003.
- [15] Sandvine. Global Internet Phenomena Spotlight: Netflix Rising. Technical report, May 2011.
- [16] T. Beauvisage. Computer Usage in Daily Life. In *Proc. ACM Computer Human Interaction (CHI)*, USA, Apr 2009.
- [17] N. Burton, M. Haynes, J. Uffelen, W. Brown, and G. Turrell. Mid-Aged Adults Sitting Time in Three Contexts. *American Journal of Preventive Medicine*, 42(4):363–373, 2012.
- [18] Broadband data block cost from Internode. http://www.internode.on.net/residential/product_features/data_blocks.
- [19] D. Rossi, M. Mellia, and M. Meo. A Detailed Measurement of Skype Network Traffic. In *Proc. USENIX IPTPS*, Tampa, Florida, USA, Feb 2008.
- [20] *flow-tools* for NetFlow. www.splintered.net/sw/flow-tools/docs/flow-tools.html.
- [21] C. Bianco, F. Cucchietti, and G. Griffa. Energy Consumption Trends in the Next Generation Access Network: A Telco Perspective. In *Proc. INTELEC*, Italy, Sep/Oct 2007.
- [22] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen. Green DSL: Energy-Efficient DSM. In *Proc. IEEE ICC*, Germany, Jun 2009.
- [23] M. Guenach, C. Nuzman, J. Maes, and M. Peeters. On Power Optimization in DSL Systems. In *Proc. GreenComm*, Germany, Jun 2009.
- [24] S. Bhaumik, D. Chuck G. Narlikar, and G. Wilfong. Energy-Efficient Design and Optimization of Wireline Access Networks. In *Proc. IEEE INFOCOM Mini-Conference*, China, Apr 2011.
- [25] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. NAPman: Network-Assisted Power Management for WiFi Devices. In *Proc. ACM MobiSys*, USA, Jun 2010.
- [26] J. Manweiler and R. R. Choudhury. Avoiding the Rush Hours: WiFi Energy Management via Traffic Isolation. In *Proc. ACM MobiSys*, USA, Jun/Jul 2011.
- [27] X. Zhang and K. G. Shen. E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks. In *Proc. ACM MOBICOM*, USA, Sep 2011.
- [28] E. Tan, L. Guo, S. Chen, and X. Zhang. CUBS: Coordinated Upload Bandwidth Sharing in Residential Networks. In *Proc. IEEE ICNP*, USA, Oct 2009.
- [29] D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo, and P. Rodriguez. Fair WLAN Backhaul Aggregation. In *Proc. ACM MOBICOM*, USA, Sep 2010.
- [30] S. Kandula, K. C.-J. Lin, T. Badirkhanli, and D. Katabi. FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput. In *Proc. USENIX NSDI*, USA, Apr 2008.
- [31] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading. In *Proc. ACM MobiSys*, USA, Jun 2007.
- [32] OpenRadio. <http://smsg.stanford.edu/projects/openradio>.
- [33] Y. Cheng et al. Automating CrossLayer Diagnosis of Enterprise Wireless Networks. In *Proc. ACM SIGCOMM*, Japan, Aug 2007.
- [34] N. Ahmed and S. Keshav. Smarta: A Self-Managing Architecture for Thin Access Points. In *Proc. ACM CoNEXT*, Portugal, Dec 2006.
- [35] R. Chandra, J. Padhye, A. Wolman, and B. Zill. A Location-Based Management System for Enterprise Wireless LANs. In *Proc. Usenix NSDI*, USA, Apr 2007.
- [36] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra. CENTAUR: Realizing the Full Potential of Centralized WLANs through a Hybrid Data Path. In *Proc. ACM MOBICOM*, China, Sep 2009.
- [37] S. Rayanchu, V. Shrivastava, S. Banerjee, and R. Chandra. FLUID: Improving Throughputs in Enterprise Wireless LANs through Flexible Channelization. In *Proc. ACM MOBICOM*, USA, Sep 2011.
- [38] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-Management in Chaotic Wireless Deployments. In *Proc. ACM MOBICOM*, Germany, Aug/Sep 2005.
- [39] A. Patro, S. Govindan, and S. Banerjee. Observing Home Wireless Experience through WiFi APs. In *Proc. ACM MOBICOM*, USA, Sep/Oct 2013.
- [40] J. Manweiler and R. R. Choudhury. RxIP: Monitoring the Health of Home Wireless Networks. In *Proc. IEEE INFOCOM*, USA, Mar 2012.



Vijay Sivaraman (M '94) received his B. Tech. degree from IIT in Delhi, India, in 1994, his M.S. from North Carolina State University in 1996, and his Ph.D. from the University of California at Los Angeles in 2000, all in Computer Science. He has worked at Bell-Labs and a silicon valley startup. He is now an Associate Professor at the University of New South Wales in Sydney, Australia. His research interests include software defined networking, and sensor networks for environmental and health applications.



John Matthews received his B.Sc. (with honours) in 1988 from the Electronic Engineering Department of Southampton University, UK. Currently, he is working as a Senior Research Projects Officer at the CSIRO Wireless and Networking Technologies Laboratory in Sydney. His expertise is in software and hardware development for network system products.



Craig Russell received his Ph.D. in Applied Mathematics from Macquarie University, Sydney in 1997. He is currently a Principal Research Engineer in the Wireless and Networking Technologies Laboratory of the CSIRO ICT Centre. He has design, implementation and operational experience in a wide range of advanced telecommunications equipment and protocols. His professional interest is the application of advanced Ethernet and IP-based technologies to Australian industries.



Syed Taha Ali did his BSc. (Eng) from GIK Institute of Engineering Sciences and Technology, Pakistan, in 2002 and his MS and PhD in Electrical Engineering from the University of New South Wales, Australia, in 2006 and 2012. His research interests include wireless sensor networks, network mobility, and security.



Arun Vishwanath (M '11) is a Research Scientist at IBM Research - Australia. He received the Ph.D. degree in Electrical Engineering from the University of New South Wales in Sydney, Australia, in 2011. He was a visiting Ph.D. scholar in the Department of Computer Science at North Carolina State University, USA in 2008. His research interests include energy-efficient networking and software defined networking.