Outsourcing Home AP Management to the Cloud through an Open API

Ashish Patro, Srinivas Govindan, Suman Banerjee
Department of Computer Sciences, University of Wisconsin Madison

1. INTRODUCTION

Networking at homes continue to get complex over time requiring users to configure and manage them. Perhaps the most common among home devices today are wireless Access Points (APs) that are supplied by a number of diverse vendors — Linksys, Netgear, DLink, TrendNet, Belkin, to name a few. In fact, wireless APs increasingly serve as the true gateway for the Internet inside a home. A plethora of WiFi-capable devices access Internet-based services through them, e.g., laptops, tablets and other handhelds, game controllers (XBox, Wii), media streaming devices (Apple TV, Google TV, Roku), and many more. Given its central role in these home networks, the performance and experience of users in the home depend centrally on efficient and dynamic configuration of these APs. In this paper, we argue for a *a simple vendor-neutral API that should be implemented by home wireless APs to enable a cloud-based management service that enables coordination, provides better performance, and reduces the burden on users.*

A cloud-based management service and a vendor-neutral API: In many dense urban environments, a large number of APs and their associated clients are in range and cause interference to each other. For example, each home AP in our deployment had 20 - 60 neighboring SSIDs . These environments are further challenged by many other wireless devices and appliances, e.g., Bluetooth headsets, analog cordless handsets, wireless security cameras, and even microwave ovens, that can also operate in the same spectrum and cause further interference. Individual home users neither have the sophistication nor the patience to frequently tune their wireless APs into some efficient configuration parameters to mitigate the impact of interference.

In our proposed service, called COAP (Coordination framework for Open APs), participating wireless APs are configured to securely connect to a cloud-based controller (Figure 1, left). The controller provides all necessary management service that can be operated by a third-party (potentially distinct from the individual ISPs). In the context of large apartment building, we envision that the apartment management contract with a single controller service (e.g., through a fixed annual fee) and all residents are asked to utilize the designated controller service within the building. This service would be no different than many other utilities distributed to residents, e.g., water, electricity, etc., which is arranged by the apartment management. Individual residents can also pick different controller services to realize many of our proposed benefits. However, some advanced features, e.g., interference management and mitigation, are better served if neighboring APs participate through the same service.

The concept of centralized management in wireless environments is not particularly new. Most enterprise WLAN solutions today (including some recent SDN-style efforts [3]) adopt this approach for managing a set of homogeneous APs in a uniform and coordinated manner. Further, a few commercial solutions, e.g., Meraki [1], provide vendor-specific proprietary cloud-managed service for APs in enterprise environments. In this paper, we argue for *an open, vendor neutral API for home APs* that should be supported by each AP manufacturer to allow third-party controller services to be designed, implemented, and deployed for *RF management of home APs*. We believe this approach is especially important in home environments where each wireless neighborhood has a diverse set of APs. Unlike enterprise environments, homogeneity in such environments is likely hard to achieve. Furthermore, with the growing demands on in-home wireless networks, especially with the dominance and growth in usage of high bandwidth media streaming, the need for coordination between neighbors will continue to grow.

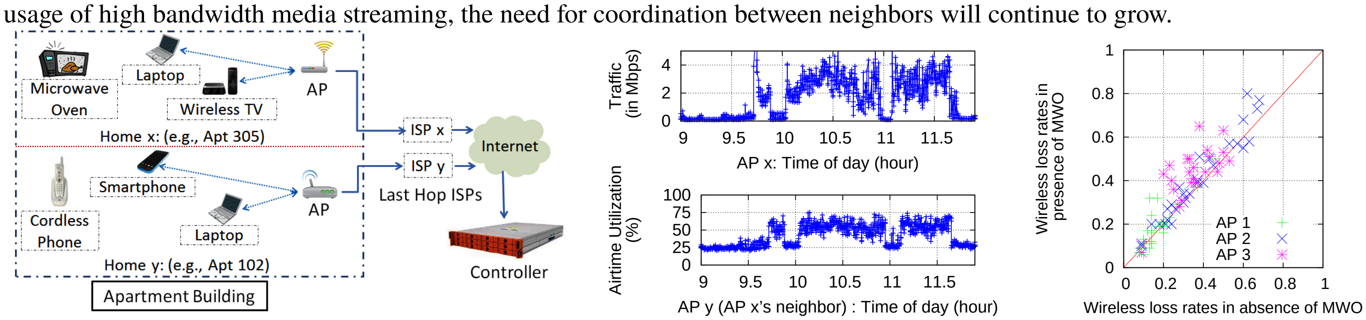

Figure 1: (Left) An example of COAP deployment within a residential apartment building consisting of COAP capable home APs and a cloud based COAP controller. (Center) WiFi traffic from AP x causing high channel utilization at AP y present in a neighboring home. (Right) Scatter plot showing per-link WiFi loss rates in the presence and absence of Microwave Ovens (MWO) activity near 3 different APs in an apartment. Points above the "y = x" line indicate the WiFi links that are potentially interfered when a neighboring Microwave Oven is active.

Function	Description
Minimal API for COAP APs — required to be implemented by every participating AP	
SetAP(channel and/or power)	Configures the AP to a particular channel and/or transmit power.
Advanced (optional) API for COAP APs — RF statistics (uses hashed version of all identifiers where needed)	
GetNeighbors()	During idle periods, the AP scans all WiFi channels for neighboring APs' beacons on each channel. Logs the each AP's MAC address (hashed) and RSSI and reports this information to the controller.
GetChannelUtil()	Records the current channel's airtime utilization over the input time duration.
GetAssociatedClients()	Get information about associated clients (e.g., MAC address (hashed), Received Signal Strength).
GetWifiStatistics()	Get packet transmission summary statistics per local link (e.g., Total packets sent, received, retried).
GetNonWifiDevices()	Report non-WiFi devices detected by the AP (type, start time, duration, RSSI), e.g., using Airshark [2].
GetPacketSummaries()	Record packet summaries for all links overheard by the AP. Each packet summary contains: received timestamp, packet length, data rate, retry bit and RSSI (average overhead < 1%).
Advanced (optional) APIs for COAP APs — flow statistics (collect specific traffic context information from APs)	
GetContext()	Report the current traffic context to the AP (e.g., traffic type such as video, VOIP, bulk download).

Table 1: The COAP API can be implemented by commodity APs. Most functions are fairly simple to implement.

Controller service and coordination: Some degree of coordination between in-range wireless transmissions is a key to improved performance. To demonstrate the benefits of a cloud-based controller service, we have been operating a preliminary version in two large apartment buildings in downtown Madison, WI. In both these apartments, we have given away free COAP WiFi APs (total of 21) to residents which have been augmented for management by our own controller service. The APs serve users in the usual manner and also provide our controller service with various measurements that demonstrate the opportunities for improvements through coordination. This system has been operational for about 5 months now, starting in September 2012 and has been observing an average of nearly 100 GB of traffic per day across all users of the system. The following are some interesting observations collected by our deployment that motivate different uses of the controller service.

A majority of observed APs (nearly 60% out of nearly 300) *never changed their channel of operation* for the entire 5 month period of the measurement study. A few of them (about 20%) changed channels only once in this entire period. In many cases this caused degraded performance to the APs and their clients. A controller service can collect observations made by one or more APs to guide channel selection to improve performance. More interesting optimizations are possible through coordination. As an example, Figure 1 (center) shows a scenario where WiFi activity started in one nearby AP that led to significant increase in channel utilization at a neighboring AP for a period of nearly 2 hours. Similarly, Figure 1 (right) illustrates how operation of microwave ovens in the vicinity of an AP (detected using Airshark [2]) increases the MAC layer loss rates of neighboring WiFi links. In each of these cases, if an AP is able to determine the *context* of its neighboring APs and clients, it can determine the best remedial measure. Individual APs, however, do not have the full view of wireless activities, since each can only observe its current channel of operation. The entire set of nearby APs can aggregate this information at the controller service, and the latter can then instruct the participating APs on potential adaptations.

2. COAP FRAMEWORK AND API

To enable COAP's usage in home settings, we propose an open API as described in Table 1. The minimum requirement for any COAP-compliant AP is to support just the basic API, that allows the controller to set the operating channel and transmit power levels. This basic functionality allows for a minimal level of control of these APs from the cloud service. Beyond this basic API, a number of advanced functionality *can be optionally* supported depending on the AP vendor's preferences. We split this optional API into two parts — those that collect anonymous wireless usage statistics, and those that leverage some information about the traffic patterns. For example, the controller can use the traffic context to anticipate future wireless usage at an AP. It is possible that an AP vendor request explicit user permission if it wants to use the traffic context part of the optional API. The implementation of each API function can be made independently by each vendor as long as the response type and format is standardized. We have been developing a specific controller service that leverages these API functions to provide a more efficient service in our initial deployments. We will continue to refine our controller implementation as well as our API specifications as we continue to learn from our deployments. The COAP framework and API is being used as an integral part of the management plane of the MobilityFirst Internet architecture (http://mobilityfirst.winlab.rutgers.edu/).

3. REFERENCES

[1] Meraki. Enterprise cloud management. http://www.meraki.com/products/wireless/enterprise-cloud-management.

[2] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: Detecting non-WiFi RF devices using commodity WiFi hardware. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11.

[3] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao. Towards programmable enterprise WLANs with Odin. HotSDN '12.

Authors are supported in part by the US National Science Foundation through awards CNS – 1258290, CNS – 1231360, CHE – 1230751, CNS – 1137558, CNS – 1064944, CNS – 1059306, CNS – 0855201, CNS – 1040648, and CNS – 0747177.