Multichannel Virtual Access Points for Seamless Handoffs in IEEE 802.11 Wireless Networks

Maria Eugenia Berezin, Franck Rousseau, Andrzej Duda
Grenoble Institute of Technology, CNRS Grenoble Informatics Laboratory UMR 5217, France
{berezin,rousseau,duda}@imag.fr

***Abstract*—Within IEEE 802.11 Wireless Local Area Networks (WLANs), client stations can move freely, but because of the short range of their Access Points (APs), they usually need to reassociate with different APs to continue to communicate. When changing APs, a client station starts a process known as a *handoff* that can take up to 2 seconds, which is too long for real-time applications such as Voice over IP (VoIP). Various solutions have been proposed to change or improve the client behaviour when doing a handoff. Previously, we proposed the idea of Virtual Access Points (VAP) implemented on APs in which a client station changes APs without disrupting its current communication. Based on this new concept, we have developed a solution called Multichannel Virtual Access Points (mVAP) to take advantage of APs operating on multiple channels. We have implemented mVAP using PACMAP, a tool for packet manipulation, and evaluated its performance. Our results show that mVAP is a new efficient technique for seamless handoffs without performance degradation.**

***Index Terms*—Wireless Local Area Networks (WLAN), IEEE 802.11, MAC layer handoff, Access Points, VoIP.**

I. Introduction

Over the past years, IEEE 802.11 Wireless Local Area Networks (WLANs) [1] have become the preferred solution to extend wired networks due to their rapid deployment and easy configuration. These characteristics, in addition to low cost hardware, have caused an increasing growth of WLANs. Wireless networking also brings the advantage of mobility allowing clients to roam freely.

We consider 802.11 infrastructure networks in which Access Points (APs) convey traffic between associated clients and the wired part of the network. Examples of such networks are university campuses, convention centers, airports, and corporation intranets. Because APs have a limited range, we can extend coverage in a larger area by deploying multiple APs, for example one AP in every office in the case of an enterprise Wi-Fi network, thus resulting in a densely deployed network. APs are interconnected through a Distribution System (DS), generally a wired network to enable inter-AP communications.

A station can join the wireless network by associating with an AP. When a station moves away from its AP, the signal of the AP falls off. If it drops below a certain threshold, the station starts searching for a new AP to associate with, initiating the MAC layer *handoff* process, until the new association takes place. During the handoff, the station neither receives nor sends data packets. This disruption in communications may take a long time that real-time applications like Voice over IP (VoIP) cannot tolerate.

Several solutions have been proposed to improve different handoff phases: discovery of new APs and reauthentication/reassociation. Most of the solutions modify the client behaviour, because the client is in charge of its association when moving and of the execution of the handoff process. As handheld VoIP WiFi devices become increasingly popular, such solutions are not practically feasible due, for example, to proprietary source code restrictions. The modification of the AP behaviour instead of clients presents thus an interesting alternative approach to improve different aspects of wireless client mobility.

In our previous work, we proposed the concept of Virtual Access Points (VAP) [2] in which a VAP manages client mobility. In this approach, a client station has the impression of always being connected to the same AP thanks to the continuous reception of beacons. Consequently, the client communication is not disrupted when changing APs, which results in better quality of service. However, all virtual APs need to operate on the same channel.

Based on this novel concept, we introduce in this paper a new solution called Multichannel Virtual Access Point (mVAP) to support seamless handoffs in networks with APs operating on multiple channels. Usually, APs in a dense WLAN use different channels to avoid interference and increase network capacity. Thus, the mVAP solution can be deployed to allow clients changing APs and channels without disrupting their communications. Clients choose new APs based on messages exchanged in the DS between APs using a specific Inter-AP Protocol. APs provide clients with the information on possible new APs so that they can change channels and continue with their connections.

The contributions of the paper are as follows:

- Proposal of a new solution called Multichannel Virtual Access Points (mVAP) for seamless and efficient handoffs.
- Implementation of the solution in a real environment using a new version of PACMAP tool running on top of the MadWifi[1] driver.
- Experimental evaluation of the mVAP performance under VoIP applications that use different voice codecs (8, 16, and 64 kbps).

In the rest of the paper, we detail the 802.11 handoff procedure and related work in Section II. Section III presents

[1] http://madwifi-project.org/

the mVAP solution. We describe its implementation in Section IV and evaluate its performance in Section V. Finally, we discuss its current limitations and possible enhancements in Section VI and conclude in Section VII.

II. Background and Related Work

A. IEEE 802.11 Handoff Procedure

When a client detects that the signal of its AP drops below a certain threshold, the client initiates the MAC layer handoff process. As described in [3], we can denote two steps in the handoff procedure:

- **Discovery:** the client searches for potential APs to associate with. It sends a *Probe Request* frame over every channel in order to know the operating APs. After sending the frame, the client listens on that channel during a *MinChannelTime* interval. If there is no answer, the channel is declared empty. On the other hand, if the client receives at least one *Probe Response* from an AP, the client waits until a *MaxChannelTime* interval to collect more information.
- **Reauthentication:** Once the client determines the new AP to associate with, the client proceeds to authenticate and, if successful, to reassociate with the new AP.

The handoff delay takes a significant amount of time: up to 2 seconds, as measured in [3], [4]. During this time, the station neither receives nor sends data packets, which may interrupt current connections. Consequently, the handoff delay is too long for real-time applications like VoIP, which recommend a one-way end-to-end delay not greater than 150 ms for good voice quality[5].

B. Related Work

Several authors proposed fast-handoff schemes to reduce the handoff delay. They fall into the following main categories: (1) reducing AP scanning (probe) time by using different strategies of channel scanning such as a proactive scan [6], a selective scan [7], eavesdropping [8], [9], and (2) reducing the authentication and reassociation time such as proactive distribution of authentication information [10].

All these schemes focus on the station behaviour, clearly because the station is the device that controls the handoff process. In our previous work, we have adopted another approach based on a technique only deployed at APs: Virtual Access Points (VAPs) [2]. Using a VAP for each station, APs control and handle a station's association state. The station has thus the impression that it is always connected to the same VAP avoiding a handoff when moving.

However, this solution presents the following limitation: all APs need to use the same channel in order to work together for client management, which results in increased interference. We therefore utilize the concept of VAPs to design a new solution for multichannel WLANs, where APs listen on different channels.

III. Multichannel Virtual Access Points

We propose to use the concept of VAP to handle seamless handoff in multichannel WLANs, where APs can listen, each one, on a different channel. Besides, using non-overlapped channels decreases interference so that the overall network capacity increases, for example channels 1, 6, and 11 in the 2.4 GHz band (802.11b/g/n).

Mobility management in Multichannel Virtual Access Points (mVAPs) using the VAP idea consists in the following procedure, as shown in Figure 1. When the station wants to join a WLAN, the AP in charge of the station's connection creates a dedicated VAP for this station. Therefore, each station associates with its owns VAP and maintains connectivity by the continuous reception of beacons. When the station moves, APs communicate between them in order to move the association state to the new AP, which now handles the VAP for the station. Therefore, the station avoids starting a handoff when moving, because it has the impression of being always connected to the same VAP, thanks to the cooperation between APs to create this virtual operation.

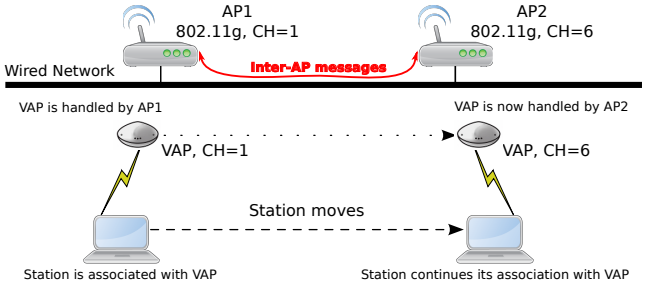


Fig. 1: Mobility management with Multichannel Virtual Access Points

As APs operate in different channels, they can cooperate through the DS commonly using an Ethernet interface. As a prerequisite, APs have information about their neighbour APs, e.g. their IP and MAC addresses. When a station moves and needs to associate with another AP and switch channels, it receives the information by means of a *Channel Switch Announcement* (CSA) element in the AP beacon. This CSA element is part of the IEEE 802.11 standard.

If there is no new AP, the station starts a standard handoff making this solution compatible with legacy 802.11 devices. This solution only requires modification of the APs behaviour, without any change on the client side.

A. Protocol details

In this section, we describe the Multichannel Virtual Access Point protocol in detail. Figure 2 presents the following steps:

1) A station (STA) is associated with AP_i on channel i. STA starts moving and AP_i detects that the signal of STA is less than a threshold *Threshold*.
2) AP_i sends to its neighbour APs ($AP_{j \neq i}$) a **Scan Request** message through the DS.
3) All $AP_{j \neq i}$ switch to channel i and listen to STA packets for a short period of time.

4) If AP_j successfully listens to STA packets, AP_j sends a **Scan Response** message to AP_i through the DS.
5) AP_i receives the **Scan Response** messages and chooses the AP with the best signal, if better than its own.
6) AP_i sends a **Station Move** message to the chosen AP_k through the DS. AP_k is listening to channel k.[2]
7) AP_k receives the **Station Move** message and starts sending the beacons for STA.
8) AP_i sends beacons to the STA with the Channel Switch Announcement (CSA) element to force the STA to switch to channel k.
9) STA receives the beacons with the CSA element and switches to channel k.
10) STA has changed AP and channel without losing connectivity. From the STA perspective, it is still connected to the same VAP.

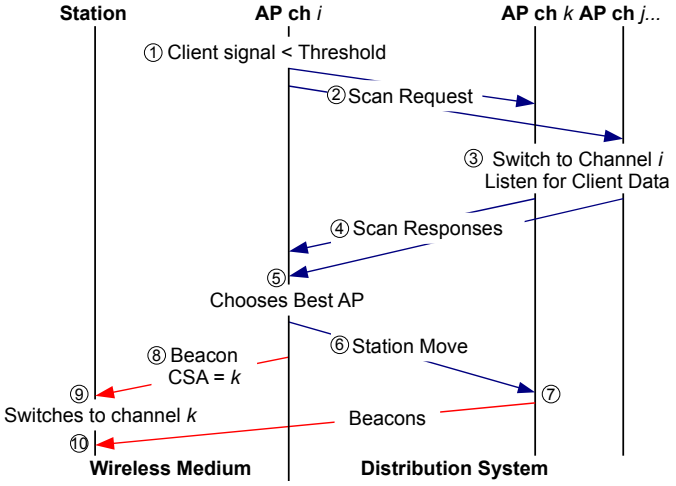


Fig. 2: Multichannel VAP protocol

IV. Implementation

We implemented the mVAP solution using our tool PACMAP: PACket MAniPulation framework. PACMAP is a framework for controlling and manipulating 802.11 frames. It is a user-space frame monitor and injector that allows for fast prototyping of modifications or customization of the IEEE 802.11 MAC protocol (management and data functions). We developed a new version of PACMAP to implement the mVAP scheme. PACMAP is currently available for download at *http://pacmap.ligforge.imag.fr/*

We have redesigned PACMAP as an event-driven C library, for Atheros-based wireless cards using the MadWifi driver. In monitor mode, the card listens to all packets and does not filter them. At the same time, MadWifi makes it possible to inject packets and send them over the wireless medium.

A TAP[3] interface emulates an Ethernet device and handles Ethernet frames. PACMAP uses a TAP interface to inject the incoming wireless data packets into the kernel network stack (i.e. the IP stack) for further processing. Inter-AP messages are exchanged using a normal TCP socket.

In our scenario, APs belong to the same subnetwork; consequently, the station can keep its IP address when changing APs. APs update bridging tables using gratuitous ARP messages.

We achieve uniqueness of VAP by including the station MAC address in the Service Set Identifier (SSID). In our example, a station with MAC address 00:11:22:33:44:55 associates to a network "Client-00:11:22:33:44:55".

[2]STA's security context could be transferred here, but we omitted this for simplicity's sake.

[3]http://vtun.sourceforge.net/tun/

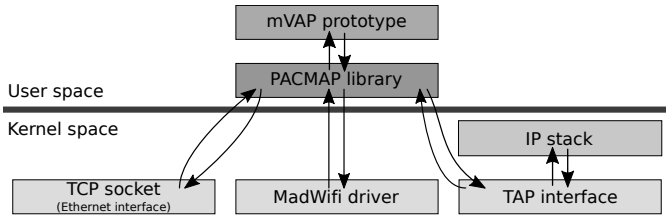


Fig. 3: PACMAP architecture and mVAP implementation

A. Inter-AP messages

Inter-AP communication takes place over the DS as an Ethernet wired network commonly interconnects APs in current deployments. Messages are sent over reliable TCP connections between APs. The Inter-AP messages contain the following information:

- **Scan Request**: station MAC address, station IP address, BSSID, station channel.
- **Scan Response**: station MAC address, station IP address, station Received Signal Strength Indicator (RSSI), new AP channel.
- **Station Move**: station MAC address, station IP address, CSA count, Beacon interval.

We have implemented Inter-AP communication using TCP sockets between the AP Ethernet interfaces. Each AP listens on a specific port and sends messages to other APs through these sockets.

B. Channel Switch Announcement

An AP uses the CSA element as described in the IEEE 802.11 standard to advertise that it is switching to a new channel. It contains the following fields:

- **Channel Switch Count**: number of beacons to listen to before channel switch. This value decreases in each consecutive beacon.
- **Channel Switch Mode**: indicates any restrictions on transmission before the channel switch.
- **New Channel Number**: the channel number after the switch.

Each station associates with its own VAP and receives a custom beacon, so the channel switch only applies to the station that changes APs. As the MadWifi driver implements the CSA element, stations can use this solution without client side modification.

The CSA element is sent in the beacon only when the new AP is chosen. In our implementation, the current AP sends 3 consecutive beacons, with an interval of 100 ms between them, decrementing in each beacon the value of Channel Switch Count. When this value reaches 0, 300 ms after the first CSA announcement, the AP deletes the station from the associated client list and stops sending beacons for the station.

V. Evaluation and Results

In this section, we first describe the setup for the experimental evaluation of the implementation of our solution mVAP and we then show its performance results.

A. Evaluation Scenario

We use two laptops acting as APs (AP1 and AP2), one laptop as a wireless mobile station (M), and one desktop computer connected to the wired network (D), as shown in Figure 4. All computers run Ubuntu 9.10. The two APs and the laptop have a D-Link wireless card with an Atheros-based chipset. The MadWifi version driver is 0.9.4. The APs run the PACMAP framework with the Multichannel VAP implementation. They use 802.11g; AP1 listens on channel 1 and AP2 listens on channel 6. The station uses the standard MadWifi driver configured in station mode.

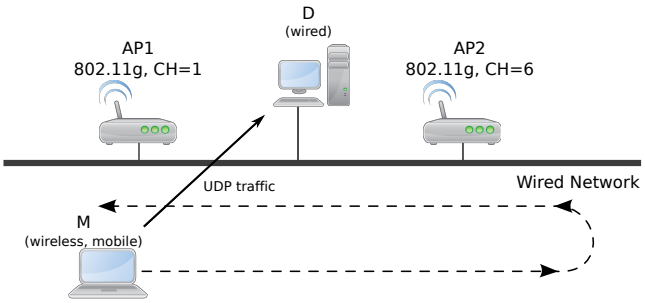


Fig. 4: Experimental setup

During the experiment, station M starts associating with *AP1* listening on channel 1 and then moves towards *AP2* listening on channel 6, that becomes its new de facto AP. Finally, station M returns to *AP1*.

For performance evaluation, we use a Constant Bit Rate (CBR) UDP stream to mimic the VoIP traffic generated by different voice codecs, shown in Table I. Packets are sent from station M to terminal D. We generate the UDP traffic with the tool *Iperf*[4]. Final UDP packet size contains the voice payload size of the codec plus 12 bytes of the Real-Time Transport Protocol (RTP) header.

B. Performance Analysis

We measure the packet Inter-Arrival Time (IAT), defined as the difference of the arrival times of the nth packet and the (n-1)th packet:

$$IAT = IAT(n) - IAT(n-1) \quad (1)$$

[4]http://iperf.sourceforge.net/

Codec	Bit Rate (Kbps)	Voice Payload Size (Bytes)	Interval (ms)
G.729	8 Kbps	20 Bytes	20 ms
G.728	16 Kbps	60 Bytes	30 ms
G.711	64 Kbps	160 Bytes	20 ms

TABLE I: Voice codecs used in the evaluation

In Figure 5, we observe in each experiment that the IAT between packets is never greater than 60 ms, thus the mVAP solution fulfils the condition of a maximum 150 ms delay for good quality of VoIP communications. Moreover, there is no packet loss when client changes AP, only one packet due to ARP table actualization. Finally, the transition from one AP to the other shows no disruption in the communication.

We present the empirical Cumulative Distributive Function of the IAT values for each codec in Figure 6. We can see that most of the values are around the expected intervals of 20 ms, 30 ms, and 20 ms for codecs G.729, G.728, and G.711, respectively. We calculate the mean, the 90 th percentile, and standard deviation of the IAT distribution, which are shown in Table II:

Codec	Mean	90th Percentile	Std. Deviation
G.729	0.02001 ms	0.02024 ms	0.00072 ms
G.728	0.03001 ms	0.03023 ms	0.00163 ms
G.711	0.02001 ms	0.02024 ms	0.00116 ms

TABLE II: Descriptors of the IAT distributions for each codec

We can conclude from these results that mVAP handles the change of AP without disrupting the current communications and offers exceptional handoff performance.

VI. Discussion

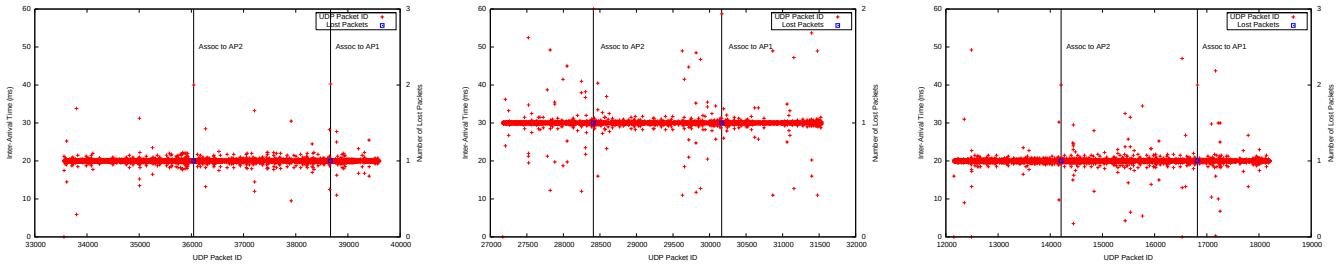
In this section we discuss the design considerations of the mVAP idea and highlight some future improvements and limitations of our solution.

A. Enhancements

One important aspect to improve is the channel switch operation. Instead of switching to another channel, APs could have another wireless card to exclusively listen to moving station packets. Thanks to this new wireless card, the AP could receive and send packets with its clients without disruption when scanning.

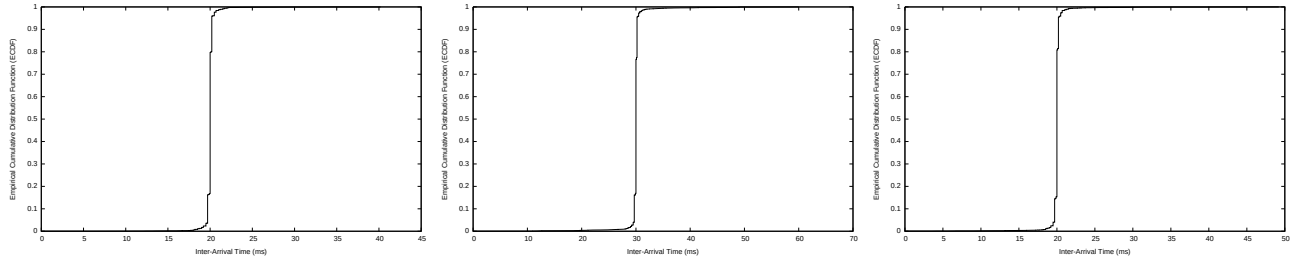
Triggering of the Scan Request and election of the best AP are currently based on instantaneous signal strength measurements. Historical information and long term trends proposed in [11] could improve these decisions using other metrics such as AP load.

Finally, authentication with the new AP should be included, as security is used in most of the current WLANs deployments. Future work should include 802.11i authentication.



(a) G.729 (8 Kbps) (b) G.728 (16 Kbps) (c) G.711 (64 Kbps)

Fig. 5: Inter-Arrival Time (IAT) between the UDP packets



(a) G.729 (8 Kbps) (b) G.728 (16 Kbps) (c) G.711 (64 Kbps)

Fig. 6: Empirical CDF of the IAT values for each codec

B. Limitations

One limitation is that collisions increase when many clients are active simultaneously, because of the overhead of handling a VAP for each station, making this solution unscalable. In order to reduce the number of beacons, APs can create VAPs as a service, only when they detect VoIP phones, for example, by distinguishing the client's MAC address from devices such as laptops.

Another limitation is that each AP needs a list of its neighbour APs in order to send the Scan Request messages. This list could be provided by an administrator, which takes an extra effort to deploy the solution, or could be generated by scanning all the channels to discover the neighbour APs.

VII. Conclusions

We have presented Multichannel VAPs, a new solution using the VAP scheme for multichannel WLANs, where the client changes AP without disrupting its current communications. The advantage of Multichannel VAPs is low latency of handoffs, which is required for multimedia applications such as VoIP. The resulting solution uses an Inter-AP protocol for communication and cooperation between APs.

We implemented the idea in a real environment, using a new version of our framework called PACMAP, in Atheros-based chipset wireless cards. PACMAP runs on top of the MadWifi driver, in user space, allowing fast development of IEEE 802.11 prototypes. We tested our implementation using three different types of voice codecs, and results showed that the delay between packets does not vary when changing AP and there is no disruption of current communications.

Finally, we discussed different aspects of this idea, such as the use of two radios, adding security, and new metrics to trigger the scanning phase and the election of the new AP. We will continue working on these points, to build a more robust solution.

Acknowledgements

This work was supported by the French National Research Agency (ANR) project ELAN under contract ANR-08-VERS-008.

References

[1] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
[2] Y. Grunenberger and F. Rousseau, "Virtual Access Points for Transparent Mobility in Wireless LANs," in *WCNC*, 2010.
[3] A. Mishra, M. Shin, and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 93–102, 2003.
[4] H. Velayos and G. Karlsson, "Techniques to reduce the ieee 802.11b handoff time," in *IEEE ICC*, 2004.
[5] International Telecommunication Union, "International Telephone Connections and International Telephone Circuits," *ITU-TG.114*, 2003.
[6] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive scan: Fast Handoff with Smart Triggers for 802.11 Wireless LAN," in *INFOCOM*, 2007.
[7] Y. Liao and L. Gao, "Practical Schemes for Smooth MAC Layer Handoff in 802.11 Wireless Networks," in *WoWMoM*, 2006.
[8] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," in *INFOCOM*, 2005.
[9] J. Teng, C. Xu, W. Jia, and D. Xuan, "D-Scan: Enabling Fast and Smooth Handoffs in AP-dense 802.11 Wireless Networks," in *INFOCOM*, 2009.
[10] A. Mishra, M. Shin, and A. W., "Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network," in *INFOCOM*, 2004.
[11] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *MobiSys*, 2006.