802.11 VLANs and Association Redirection

Jon Ellch

Contents

Chapter 1

Foreword

Abstract: The goal of this paper is to introduce the reader to association redirection and how it could to used to implement something analogous to VLANs found in wired media into a typical IEEE 802.11 environment. What makes this technique interesting is that it can be accomplished without breaking the IEEE 802.11 standard on the client side, and requires only minor changes made to the Access Point (AP). No modifications are made to the 802.11 MAC. It is the author's hope that after reading this paper the reader will not only understand the specific technique outlined below, but will consider protocol quirks with a new perspective in the future.

Chapter 2

Background

The IEEE 802.11 specification defines a hierarchy of three states a client can be in. When a client wishes to connect to an Access Point (AP) he progresses from state 1 to 2 to 3. The client progresses initially from state 1 to state 2 by successfully authenticating (this authentication stage happens even when there is no security enabled). Similarly the client progresses from state 2 to 3 by associating. Once a client as associated he enters state 3 and can transmit data using the AP.

Unlike ethernet, 802.3, or other link layer headers, 802.11 headers contain at least 3 addresses: source, destination, and Basic Service Set ID (BSSID). The BSSID can be best thought of as a through field. Packets destined for the APs interface have both destination and BSSID set to the same value. A packet destined to a different host on the same WLAN however would have the BSSID set to the AP and the destination set to the host.

The state transition diagram in the standard dictates that if a client receives an association response with a different source address than it was expecting, then the client should set his BSSID to the new source address. The technique of sending an association response with a different source address is known as *association redirection*. While the motivation for this idiosyncrasy is unclear, it can be leveraged to dynamically create what has been described as a `personal virtual bridged LAN (PVLAN)`.

Chapter 3

Introduction

The most compelling reason to virtualize APs has been security. There are currently two possible techniques for doing this, though only one has been deployed in the wild. The most prevalent has been implemented by Colubris in their virtual access point technology [4].

The other technique, public access point (PAP) and personal virtual bridged LANs (PVLANs), which is described in this paper, has been documented in U.S. patent no. 20040141617[1].

3.1 The state of the art

The Colubris virtual access point technology is a single physical device that implements an entirely independent 802.11 MAC protocol layer (including a unique BSSID) for each virtual AP. The only thing shared between the individual virtual APs is the hardware they are running on. The device goes so far as to implement virtual Management Information Bases (MIBs) for each virtual AP. The Colubris solution fits well into a heavily managed static environment where the users and the groups they belong to are well defined. Deploying it requires that each user knows which SSID to associate with a priori, along with any required authentication credentials. The virtual access point is capable of mapping virtual access points into 802.1q VLANs.

The public AP solution fits well into less managed networks. Public AP utilizes the technique outlined in this paper. The Public AP broadcasts a single beacon for a Public Access Point (PAP). When a client attempts to associate, the PAP redirects him to a dynamically generated VBSSID, placing him on his own PVLAN. This is well suited to a typical hotspot scenario where there is no implicit trust between users, and the number of clients is not known beforehand. This technique could also be used in conjunction with traditional 802.1q VLANs, however its strength lies in the lower burden of administrative requirements. This technique is designed to work well when deployed in the common hot spot scenario where the administrators have little other network infrastructure and the only thing upstream is a best effort common carrier provider.

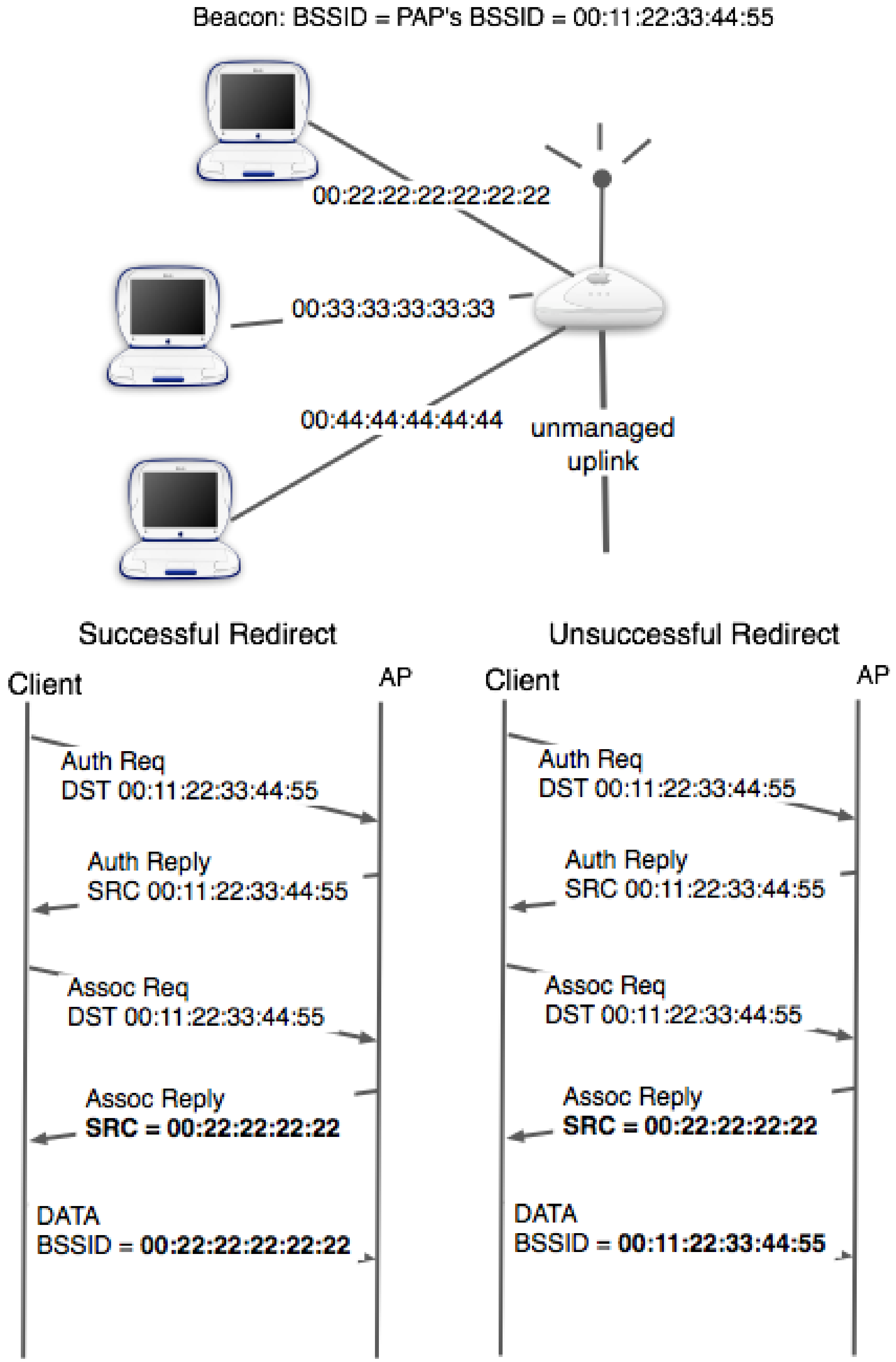
Chapter 4

PVLANs and virtual BSSIDs

PVLANs are called Personal Bridged VLANs because the VLAN is created dynamically for the client. The client essentially owns the VLAN since he controls its creation and its lifetime. In the most common scenario there would only be a single client per PVLAN.

An access point that implements the PAP concept intentionally re-directs associating clients to their own dynamically generated BSSID (Virtual BSSID or VBSSID).

In the example below the AP is broadcasting a public BSSID of 00:11:22:33:44:55 and is redirecting the client to his own VBSSID 00:22:22:22:22:22.


Beacon: BSSID = PAP's BSSID = 00:11:22:33:44:55
00:22:22:22:22:22:22
00:33:33:33:33:33
00:44:44:44:44:44
unmanaged
uplink
Successful Redirect
Client
AP
Auth Req
DST 00:11:22:33:44:55
Auth Reply
SRC 00:11:22:33:44:55
Assoc Req
DST 00:11:22:33:44:55
Assoc Reply
SRC = 00:22:22:22:22
DATA
BSSID = 00:22:22:22:22:22
Unsuccessful Redirect
Client
AP
Auth Req
DST 00:11:22:33:44:55
Auth Reply
SRC 00:11:22:33:44:55
Assoc Req
DST 00:11:22:33:44:55
Assoc Reply
SRC = 00:22:22:22:22
DATA
BSSID = 00:11:22:33:44:55

Chapter 5

The Experiment

The experiment conducted was not a full-blown implementation of a PAP. The experiment was designed to test a wide variety of chipsets, cards, and drivers for compatibility with the standard and susceptibility to association re-direction. To this end all the cards were subjected to every reasonable intrepretation of the standard.

The experiment was conducted by making some simple changes to the host-ap driver on Linux. Host-ap can operate in Access Point mode as well as in client mode. All the modifications were made in Access Point mode. Host-ap's client-side performance is unrelated to the changes made for the experiment.

The experiment was conducted in two phases. First, host-ap was modified to mangle all management frames by modifying the source, BSSID, source and BSSID (at the same time). The results of this are reflected in table one.

After this was complete, host-ap was modified to return authentication replies un-mangled. This was due to the amount of cards that simply ignored mangled authentication replys. These results are cataloged in table two.

5.1 The Results

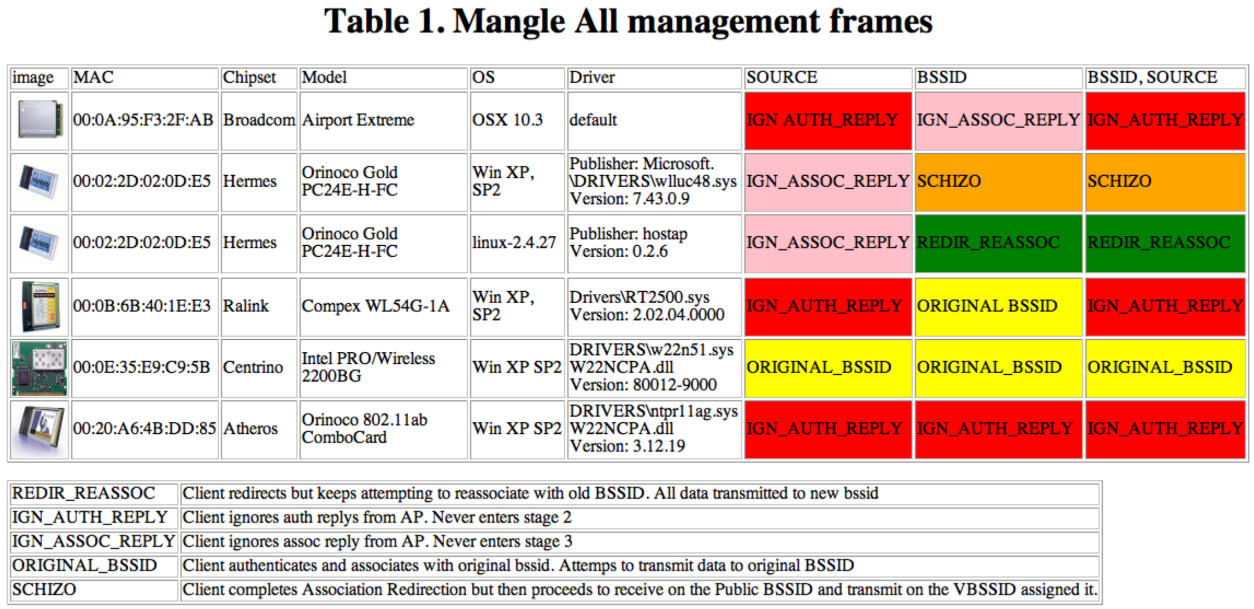


Table 1. Mangle All management frames

image	MAC	Chipset	Model	OS	Driver	SOURCE	BSSID	BSSID, SOURCE
	00:0A:95:F3:2F:AB	Broadcom	Airport Extreme	OSX 10.3	default	IGN AUTH_REPLY	IGN_ASSOC_REPLY	IGN_AUTH_REPLY
	00:02:2D:02:0D:E5	Hermes	Orinoco Gold PC24E-H-FC	Win XP, SP2	Publisher: Microsoft. \DRIVERS\wlluc48.sys Version: 7.43.0.9	IGN_ASSOC_REPLY	SCHIZO	SCHIZO
	00:02:2D:02:0D:E5	Hermes	Orinoco Gold PC24E-H-FC	linux-2.4.27	Publisher: hostap Version: 0.2.6	IGN_ASSOC_REPLY	REDIR_REASSOC	REDIR_REASSOC
	00:0B:6B:40:1E:E3	Ralink	Compex WL54G-1A	Win XP, SP2	Drivers\RT2500.sys Version: 2.02.04.0000	IGN_AUTH_REPLY	ORIGINAL BSSID	IGN_AUTH_REPLY
	00:0E:35:E9:C9:5B	Centrino	Intel PRO/Wireless 2200BG	Win XP SP2	DRIVERS\w22n51.sys W22NCPA.dll Version: 80012-9000	ORIGINAL_BSSID	ORIGINAL_BSSID	ORIGINAL_BSSID
	00:20:A6:4B:DD:85	Atheros	Orinoco 802.11ab ComboCard	Win XP SP2	DRIVERS\ntpr11ag.sys W22NCPA.dll Version: 3.12.19	IGN_AUTH_REPLY	IGN_AUTH_REPLY	IGN_AUTH_REPLY

REDIR_REASSOC	Client redirects but keeps attempting to reassociate with old BSSID. All data transmitted to new bssid
IGN_AUTH_REPLY	Client ignores auth replys from AP. Never enters stage 2
IGN_ASSOC_REPLY	Client ignores assoc reply from AP. Never enters stage 3
ORIGINAL_BSSID	Client authenticates and associates with original bssid. Attemps to transmit data to original BSSID
SCHIZO	Client completes Association Redirection but then proceeds to receive on the Public BSSID and transmit on the VBSSID assigned it.

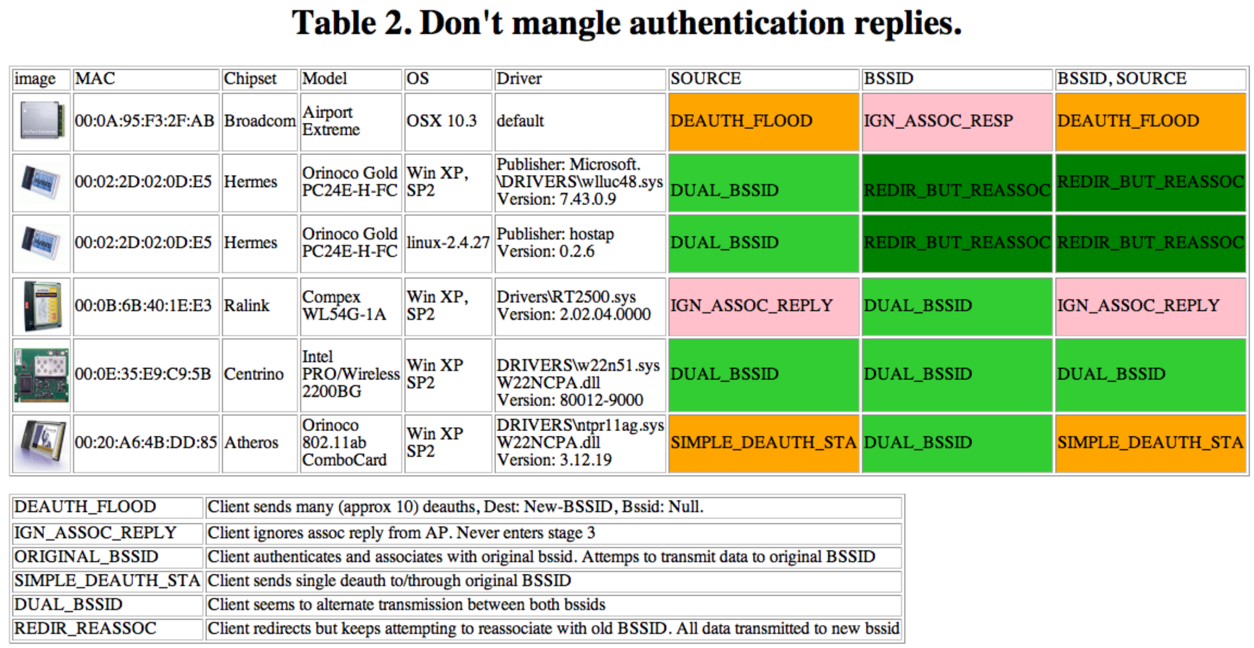


Table 2. Don't mangle authentication replies.

image	MAC	Chipset	Model	OS	Driver	SOURCE	BSSID	BSSID, SOURCE
	00:0A:95:F3:2F:AB	Broadcom	Airport Extreme	OSX 10.3	default	DEAUTH_FLOOD	IGN_ASSOC_RESP	DEAUTH_FLOOD
	00:02:2D:02:0D:E5	Hermes	Orinoco Gold PC24E-H-FC	Win XP, SP2	Publisher: Microsoft. \DRIVERS\wlluc48.sys Version: 7.43.0.9	DUAL_BSSID	REDIR_BUT_REASSOC	REDIR_BUT_REASSOC
	00:02:2D:02:0D:E5	Hermes	Orinoco Gold PC24E-H-FC	linux-2.4.27	Publisher: hostap Version: 0.2.6	DUAL_BSSID	REDIR_BUT_REASSOC	REDIR_BUT_REASSOC
	00:0B:6B:40:1E:E3	Ralink	Compex WL54G-1A	Win XP, SP2	Drivers\RT2500.sys Version: 2.02.04.0000	IGN_ASSOC_REPLY	DUAL_BSSID	IGN_ASSOC_REPLY
	00:0E:35:E9:C9:5B	Centrino	Intel PRO/Wireless 2200BG	Win XP SP2	DRIVERS\w22n51.sys W22NCPA.dll Version: 80012-9000	DUAL_BSSID	DUAL_BSSID	DUAL_BSSID
	00:20:A6:4B:DD:85	Atheros	Orinoco 802.11ab ComboCard	Win XP SP2	DRIVERS\ntpr11ag.sys W22NCPA.dll Version: 3.12.19	SIMPLE_DEAUTH_STA	DUAL_BSSID	SIMPLE_DEAUTH_STA

DEAUTH_FLOOD	Client sends many (approx 10) deauths, Dest: New-BSSID, Bssid: Null.
IGN_ASSOC_REPLY	Client ignores assoc reply from AP. Never enters stage 3
ORIGINAL_BSSID	Client authenticates and associates with original bssid. Attemps to transmit data to original BSSID
SIMPLE_DEAUTH_STA	Client sends single deauth to/through original BSSID
DUAL_BSSID	Client seems to alternate transmission between both bssids
REDIR_REASSOC	Client redirects but keeps attempting to reassociate with old BSSID. All data transmitted to new bssid

The responses in table one varied all the way from never leaving stage 1 to successful redirection. The most interesting cases are the drivers that successfully made it to stage 3. There are three cases of this. The cases marked ORIGINAL_BSSID are what was initially expected from many devices, that they would simply ignore the redirect request and continue to transmit on the PAP BSSID. The REDIRECT_REASSOC case is a successful redirection with a small twist. The card transmits all data to VBSSID, however it periodically sends out reassociation requests to the PAP BSSID.

The SCHIZO case is the other case that made it into stage 3. In this case the card is listening on the PAP BSSID and then proceeds to transmit on the VBSSID. The device seems to ignore any data transmitted to it on the VBSSID.

As mentioned previously in table two, the possibilty of ignoring authentication reply's has been eliminated by not mangling fields until the association request. This opened up the possibilty for some interesting responses.

The Apple airport extreme card responded with a flood of deauthentication packets to the null BSSID with a destination of the AP (DEAUTH_FLOOD). The Atheros card is the only other card that sent a deauth, though it had a much more measured response, sending a single de-auth to the original BSSID (SIMPLE_DEAUTH_STA).

The other new response in table 2 is the DUAL_BSSID behavior. These cards seem to alternate intentionally between both BSSIDS on every other transmitted packet. It is unknown whether they continue to do this for the entire connection or if this is some sort of intentional behavior and they will choose whichever BSSID they receive data on first.

The experiment provided some very surprising results. Originaly it was suspected that many cards would simply never enter stage 3, or alternately just use the original BSSID they set out to. Quite a few cards can be convinced to go into dual BSSID behavior and might be susceptible to association redirection. Two drivers for the hermes chipset were successfuly redirected.

Chapter 6

Future Work

Clearly modifying client side drivers for better standards compliance is one area work could be done. More interesting questions are how does one handle key management on the AP in this situation? Clearly any PSK solutions don't really apply in this scenario. How much deviation from the spec needs to happen for WPA 802.1x authentication to successfully be deployed? One interesting area of research is the concept of a stealthy rogue AP.

By using association redirection clients could be the victim of stealthy (from the perspective of the network admin) association hijacking from a rogue AP. An adversary could just set up shop with a modified host-ap driver on a Linux box that didn't transmit beacons. Rather it would wait for a client to attempt an association request with the legitimate access point and try to win a race condition to see who could send an association reply first. Alternately the adversary could simply de-authenticate the user and then be poised to win the race.

Another interesting question is the whether or not a PAP could withstand a DOS attack attempting to create an overwhelming amount of VBSSIDs. It is the authors opinion that a suitable algorithm could be found to make the resources required for the attack too costly for most. By dynamically expiring PVLANs and VBSSIDs as a function of time and traffic the PAP could burden the attacker with keeping track of all his VBSSIDs as well, instead of just creating as many as he can and forgetting about them.

Chapter 7

Conclusion

It is unlikely that this technique could be successfully be deployed to create PVLAN's in a general scenario due to varied behavior from the vendors. However, it does appear that a determined attacker could encode the data generated from this experiment into a modified host-ap driver so that he could stealthily redirect traffic to himself. This would give the attacker a slight advantage over typical ARP poisioning attacks since he doesn't need to generate any suspicous ARP activity. It also has an advantage over simple rogue access points, as it requires no beacons which can easily be detected.

Bibliography

[1] Volpano, Dennis. *United States Patent Application 200403141617* July 22, 2003
http://appft1.uspto.gov/netahtml/PTO/search-adv.html

[2] Institute of Electrical and Electronics Engineers. *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11-1999, 1999. (pg 376)* 1999

[3] Aboba, Bernard. *Virtual Access Points (IEEE document IEEE 802.11-03/154r1)* May 22, 2003
http://www.drizzle.com/~aboba/IEEE/11-03-154r1-I-Virtual-Access-Points.doc

[4] Colubris Networks. *Virtual Access Point Technology Multiple WLAN Services*
http://www.colubris.com/literature/whitepapers.asp
accessed Aug 09, 2005.