Setting up Linux-based Access Points with HostAP

Air Bridge

HostAP is a software that makes a flexible and powerful WLAN Access Point of a Linux computer. This article looks into the various configuration options, focusing on an example based on Ethernet bridging and roaming clients.

BY CORINNA AICHELE

Fancy a mesh router in your attic with an integrated WLAN base station to allow your fellow residents and neighbors access to the local Community mesh network? How about adding an Access Point to your home router? All you need to build a Linux-based Access Point that is far superior to an out-of-the-box base station is an old computer and a suitable WLAN card.

Some older Access Points use normal PCMCIA cards and a firmware to provide base station functionality. The HostAP project has developed a driver that allows WLAN cards with the low-budget Intersil Prism-2/2.5/3 chipset to provide this functionality.

In terms of memory and storage, a six-year old PC is powerful in comparison to an Access Point – although the power consumption is admittedly a lot higher. Free software is available for Linux to support features such as Quality-of-Service routing.

Supported WLAN Cards are Becoming Rare

Unfortunately, cards with supported Prism chipsets are slowly but surely disappearing from the shops. Also, some manufacturers change the chipsets on well-known models without informing potential purchasers. Just a few months ago, most low-budget 11 MBit cards were based on the Prism 2/2.5/3 chip by Intersil. Sadly, these chipsets have been replaced in 11 MBit cards with chips by Atmel or the cheaper Realtek 8180L chip.

Atmel-based cards work quite well under Linux, but they are not suitable for use as Access Points. Realtek are rumored to work under Linux, and as base stations using the latest binary open source driver, however, I was unable to confirm this.

The Netgear MA401 card for laptops, and the MA301 PCI card, are still available and based on a suitable Prism chipset. Allnet also has suitable PCMCIA and PCI cards, including a PCMCIA model with an external antenna port. Most PCMCIA cards with Prism chipsets are missing the antenna connector which is typical of PCI cards. This is a big disadvantage for a base station.

Home Brew Tastes Better

Two Linux variants dedicated to building WLAN routers include the HostAP driver by default. These are Cqureap [1], which fits on a single floppy, and Mesh Linux [2], which is designed for hard disk installation. The HostAP driver is not included by most major distros – with the exception of Debian and Mandrake 9 – there are unofficial RPMs for Red Hat 8 and 9, although they use the obsolete 0.0.3 version [3]. The version included by Mandrake 9.1 and 9.2 is more recent, 0.0.4, but still not the latest.

Listing 1: AP-A Configuration

```
01 iwpriv wlan0 wds_add
   00:22:22:22:22:22
02 brctl addbr br0
03 brctl addif br0 eth0
04 brctl addif br0 wlan0
05 brctl addif br0 wlan0wds0
06 ifconfig eth0 0.0.0.0
07 ifconfig wlan0 0.0.0.0
08 ifconfig wlan0wds0 0.0.0.0
09 ifconfig br0 172.16.0.1 up
```

It makes sense to build the current version 0.2.4 of the HostAP driver yourself. For PCMCIA cards you will also need the source code for the external PCMCIA card service, if you are not using kernel-based PCMCIA support. To install from the source code, you need to download the gzipped tar archive, *hostap-driver-xx.tar.gz*, from the developer website [4], and unpack with *tar xvzf*, as usual.

The kernel sources need to be installed for you to compile the source code. The kernel sources are typically stored in a subdirectory below */usr/src*, and a symlink in */usr/src/linux* usually points to them. If this is not the case, you need to modify the *KERNEL_PATH = /usr/src/linux* line in the *Makefile*.

External & Kernel-Based PCMCIA

To use external PCMCIA card services, first rename or copy the Makefile:

```
mv pcmcia-cs-xx/driver /modules
/Makefile pcmcia-cs-xx/ driver
/modules/Makefile-Safecopy
```

Then copy the *hostap-driver-xx/driver/* directory to *pcmcia-cs-versionnumber/driver/*, build and install pcmcia-cs:

```
cp -a hostap-driver-xx/driver/*
 /usr/src/pcmcia-cs-xx/driver
make config
make &&
make install
```

Relaunch the card manager to apply the changes. The syntax for this command depends on the init script style, but it is typically:

```
etc/init.d/pcmcia restart
```

or

```
etc/rc.d/rc.pcmcia restart
```

Kernel-based PCMCIA is a lot easier to configure. If the path to the kernel sources in the Makefile is correct, simply call *make && make install* in the *hostap-driver-xx/* directory and launch the card manager.

The wireless tools, including *iwconfig* and *iwpriv*, control the WLAN interface. The command for enabling Access Point mode in *iwconfig* is *mode master*:

```
iwconfig wlan0 essid linux-➥
magazine channel 10 mode master
```

Before you can use WEP encryption, a typical characteristic of WLAN cards, you need to install the *hostap_crypt_wep* module by entering *modprobe hostap_crypt_wep*. This allows you to specify the WEP key:

```
iwconfig wlan0 key s:password
```

MAC Based Access Filters

You can use MAC addresses to allow or deny network access. The following commands allow only the WLAN cards in the list to access the wireless network:

```
iwpriv wlan0 maccmd 1
iwpriv wlan0 addmac ➥
00:11:11:11:11:11
iwpriv wlan0 addmac ➥
00:33:33:33:33:33
```

The following commands deny access to the specified MAC addresses, although this does not make much sense, as MAC addresses can be spoofed with a little help from *ifconfig*:

```
iwpriv wlan0 maccmd 2
iwpriv wlan0 addmac ➥
00:11:22:33:44:55
iwpriv wlan0 addmac ➥
00:11:33:22:33:11
```

The HostAP driver not only supports Access Point mode, but also ad-hoc, managed client, WDS (Wireless Distribution System), and monitor mode (used for WarDriving).

Listing 2: AP-B Configuration

```
01 iwpriv wlan0 wds_add
   00:11:11:11:11:11
02 brctl addbr br0
03 brctl addif br0 eth0
04 brctl addif br0 wlan0wds0
05 ifconfig eth0 0.0.0.0
06 ifconfig wlan0wds0 0.0.0.0
07 ifconfig br0 172.16.0.2 up
```

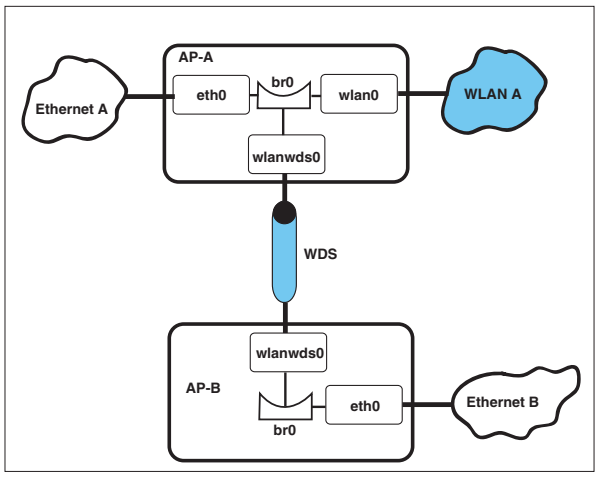

Figure 1: The WLAN bridges two Ethernet networks. To allow this to happen, HostAP needs to be running in WDS mode. Access Point A also allows WLAN clients to associate.

Wireless Distribution System

WDS allows you to combine multiple Access Points and clients to roam between APs. Access Points in WDS mode can also act as bridges between two segments of a wired network. However, WDS requires a Prism firmware 1.5.0 or newer. Tip: The *hostap-utils-xx.tar.gz* on the developer website [1] also has the source code for a tool with which you can flash the card firmware. You need to install the *bridge-utils* for bridge management [5]. They are included with most major distributions.

Bridge Building

For our example, we will be using two Host APs to build a bridge between two segments of a wired Ethernet network. WLAN clients can associate with the base station and roam between the three subnets – Ethernet A, Ethernet B, and Wireless LAN (WDS). The driver automatically detects the subject the client is on. Our example uses a private Class B network address, 172.16.0.0/16. We will be using the same network addresses and netmasks in all three segments. As the interfaces build a layer 2 bridge, the whole network looks like a single physical network to the devices on it (see Figure 1).

The interface to the wired Ethernet is *eth0* for both APs, and *wlan0* is the WLAN interface used by HostAP. The WLAN card in base station AP-A has a MAC address of *00:11:11:11:11:11*, and an IP of *172.16.0.1*. The WLAN card in base station AP-B has an address of *00:22:22:22:22:22* and an IP of *172.16.0.2*.

In this special case, the *wlan0* interface in Access Point AP-B is not used. All wireless traffic is routed by WDS to Access Point AP-A via the *wlan0wds0* interface. The commands for configuring both Access Points are shown in Listings 1 and 2. The *brctl show* command shows the characteristics of the bridge, *br0*. To remove the need to re-configure each time you reboot, you might like to add a shell script, and launch the script at the appropriate runlevel.

Don't Forget to Encrypt!

If you intend to run the AP in a production environment, you should use IPSec for traffic encryption. The readme file supplied with the HostAP driver has information on that topic, and on the configuration options. This is also where our example comes from. Check out the *waproamd* and *ifplugd* [6] projects, for dynamic client-side configuration. Have fun trying out HostAP! ■

INFO

[1] Cqureap Linux router on a floppy: *http://freshmeat.net/projects/cqureap/*

[2] Mesh Linux: *http://scii.nl/~elektra/*

[3] HostAP RPMs for Red Hat and Suse kernels: *http://atrpms.net/dist/rhg/hostap-driver/*

[4] HostAP Project: *http://hostap.epitest.fi/*

[5] Bridge tools: *http://bridge.sourceforge.net/*

[6] *waproamd* and *ifplugd*: *http://opointer.de/lennart/*