

NAME

wlan - generic WiFi 802.11 link-layer support

SYNOPSIS

device wlan

DESCRIPTION

The **wlan** module provides generic code to support 802.11 drivers. Where a device does not directly support 802.11 functionality this layer fills in. The **wlan** module is required by all native 802.11 drivers.

wlan supports multi-mode devices capable of operating in both 2.4GHz and 5GHz bands and supports numerous 802.11 standards: 802.11a, 802.11b, 802.11g, 802.11n, and 802.11s (Draft 3.0). The WPA, 802.11i, and 802.1x security protocols are supported through a combination of in-kernel code and user-mode applications. The WME/WMM multi-media protocols are supported entirely within the **wlan** module but require a suitably capable hardware device. Likewise the 802.11h specification is supported only by suitably capable devices.

Drivers provide 802.11 functionality through **wlan** interfaces that are created at runtime using interface cloning. This is done with the `ifconfig(8)` **create** command or using the `wlans_IFX` variable in `rc.conf(5)`. Some drivers support the creation of multiple **wlan** interfaces that share the same underlying device; this is the way by which “multi-bss support” is provided but it can also be used to create WDS links and other interesting applications.

There are several types of **wlan** interfaces that may be created:

sta A client station in an infrastructure bss (i.e. one that associates to an access point).

hostap An access point in an infrastructure bss.

mesh A mesh station in an MBSS network.

adhoc A station in an IBSS network.

ahdemo A station operating in “adhoc demo mode”. This is essentially an IBSS station that does not use management frames (e.g. no beacons are transmitted). An **ahdemo** interface is especially useful for applications that want to transmit and receive raw 802.11 packets.

monitor An interface used exclusively for capturing 802.11 frames. In particular this is specified to have read-only properties which enables it to be operated on frequencies where one would otherwise not be allowed.

wds A station that passes 4-address 802.11 traffic for the purpose of tunneling traffic over a wireless link. Typically this station would share the same MAC address as a **hostap** interface. It may be possible to create **wds** interfaces without a companion **hostap** interface but that is not guaranteed; one may need to create a **hostap** interface that does not send beacon frames before **wds** interfaces may be created.

Note that an interface's type cannot be changed once it is created.

wlan defines several mechanisms by which plugin modules may be used to extend its functionality. Cryptographic support such as WEP, TKIP, and AES-CCMP are implemented as standalone modules (if not statically configured into a system) that register with **wlan**. Similarly there is an authenticator framework for defining 802.11 authentication services and a framework for integrating access control mechanisms specific to the 802.11 protocol.

DEBUGGING

If the IEEE80211_DEBUG option is included in the kernel configuration, debugging controls are available using:

```
sysctl net.wlan.X.debug=mask
```

where *X* is the number of the **wlan** instance and *mask* is a bit-or of control bits that determine which debugging messages to enable. For example,

```
sysctl net.wlan.0.debug=0x00200000
```

enables debugging messages related to scanning for an access point, adhoc neighbor, or an unoccupied channel when operation as an access point. The `wldebug(8)` tool provides a more user-friendly mechanism for doing the same thing. Note that

```
sysctl net.wlan.debug=mask
```

defines the initial value of the debugging flags for each cloned **wlan** interface; this is useful to enable debug messages during interface creation.

COMPATIBILITY

The module name of **wlan** was used to be compatible with NetBSD.

Mesh stations follow the 802.11s Draft 3.0 specification which is not ratified and subject to change. Be aware that this specification is incompatible with earlier drafts. Stations implementing earlier drafts (e.g., Linux) may be incompatible.

SEE ALSO

ath(4), bwi(4), bwn(4), ipw(4), iwi(4), iwlwifi(4), iwm(4), iwn(4), malo(4), mwl(4), netintro(4), otus(4), ral(4), rsu(4), rtw88(4), rtwn(4), rum(4), run(4), uath(4), upgt(4), ural(4), urtw(4), wlan_acl(4), wlan_ccmp(4), wlan_tkip(4), wlan_wep(4), wlan_xauth(4), wpi(4), zyd(4)

STANDARDS

More information can be found in the IEEE 802.11 Standards.

HISTORY

The **wlan** driver first appeared in FreeBSD 5.0.

AUTHORS

Atsushi Onoe is the author of original NetBSD software from which this work began. Sam Leffler brought the code into FreeBSD and then rewrote it to support multi-mode devices, 802.11g, 802.11n, WPA/802.11i, WME, multi-bss, and add the extensible frameworks for cryptographic, authentication, and access control plugins. This manual page was written by Tom Rhodes <trhodes@FreeBSD.org>.