NAME

    wpi - Intel 3945ABG Wireless LAN IEEE 802.11 driver

SYNOPSIS

    To compile this driver into the kernel, place the following lines in your kernel configuration file:

        device wpi
        device wpifw
        device pci
        device wlan
        device wlan_amrr
        device firmware

    Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

        if_wpi_load="YES"

DESCRIPTION

    The **wpi** driver provides support for the Intel 3945ABG Wireless network adapter. The driver supports **station**, **adhoc**, **adhoc-demo**, **hostap**, and **monitor** mode operation. Only one virtual interface may be configured at any time.

    The **wpi** driver can be configured to use Wired Equivalent Privacy (WEP) or Wi-Fi Protected Access (WPA-PSK and WPA2-PSK). WPA is the de facto encryption standard for wireless networks. It is strongly recommended that WEP not be used as the sole mechanism to secure wireless communication, due to serious weaknesses in it. The **wpi** driver offloads both encryption and decryption of data frames to the hardware for the CCMP cipher.

    This driver requires the firmware built with the **wpifw 4** module to work.

    The **wpi** driver can be configured at runtime with ifconfig(8).

FILES

    /usr/share/doc/legal/intel_wpi.LICENSE **wpi** firmware license

EXAMPLES

    Join an existing BSS network (i.e., connect to an access point):

        ifconfig wlan0 create wlandev wpi0 inet 192.168.0.20 \
            netmask 0xffffff00

Join a specific BSS network with network name "my_net":

    ifconfig wlan0 create wlandev wpi0 ssid my_net up

Join a specific BSS network with 64-bit WEP encryption:

    ifconfig wlan0 create wlandev wpi0 ssid my_net \
            wepmode on wepkey 0x1234567890 weptxkey 1 up

Create an IBSS network with 128-bit WEP encryption on the channel 4:

    ifconfig wlan0 create wlandev wpi0 wlanmode adhoc ssid my_net \
            wepmode on wepkey 0x0102030405060708091011213 weptxkey 1 \
            channel 4

Join/create an 802.11b IBSS network with network name "my_net":

    ifconfig wlan0 create wlandev wpi0 wlanmode adhoc
    ifconfig wlan0 inet 192.168.0.22 netmask 0xffffff00 ssid my_net \
            mode 11b

Create an 802.11g host-based access point:

    ifconfig wlan0 create wlandev wpi0 wlanmode hostap
    ifconfig wlan0 inet 192.168.0.10 netmask 0xffffff00 ssid my_ap \
            mode 11g

DIAGNOSTICS

wpi%d: could not load firmware image '%s' The driver failed to load the firmware image using the firmware(9) subsystem. Verify the wpifw(4) firmware module is installed.

wpi%d: %s: timeout waiting for adapter to initialize, error %d The onboard microcontroller failed to initialize in time. This should not happen.

wpi%d: %s: could not load boot firmware An attempt to upload the boot firmware image to the onboard microcontroller failed. This should not happen.

wpi%d: device timeout A frame dispatched to the hardware for transmission did not complete in time. The driver will reset the hardware and continue. This should not happen.

wpi%d: scan timeout Firmware scan command response was not received in time. The driver will reset the hardware and continue. This should not happen.

wpi%d: fatal firmware error The onboard microcontroller crashed for some reason. The driver will reset the hardware and continue. This should not happen.

wpi%d: RF switch: radio disabled The hardware switch controlling the radio is currently turned off. Data transmission is not possible in this state.

wpi%d: can't map mem space The driver was unable to map the device registers into the host address space. This should not happen.

wpi%d: can't map interrupt The driver was unable to allocate an IRQ for the device interrupt. This should not happen.

wpi%d: can't establish interrupt, error %d The driver was unable to install the device interrupt handler. This should not happen.

wpi%d: %s: bus_dmamap_load failed, error %d The driver was unable to map newly allocated mbuf to device visible address space. Contents of currently received frame will be lost. This should not happen.

SEE ALSO

pci(4), wlan(4), wlan_amrr(4), wlan_ccmp(4), wlan_tkip(4), wlan_wep(4), wlan_xauth(4), wpifw(4), hostapd(8), ifconfig(8), wpa_supplicant(8)

AUTHORS

The original **wpi** driver was written for OpenBSD by Damien Bergamini *<damien.bergamini@free.fr>*. Benjamin Close *<benjsc@FreeBSD.org>* ported **wpi** to FreeBSD.

CAVEATS

Hostap mode is not directly supported by the device; it is implemented through IBSS mode (as a result, DFS/passive channels are not available in this mode).

Powersave may be unstable on some networks (results in occasional 'wpi%d: device timeout' messages); you can try to disable it to improve device stability.