NAME

rtwn - Realtek IEEE 802.11 wireless network driver

SYNOPSIS

options RTWN_DEBUG
options RTWN_WITHOUT_UCODE

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

```
device rtwn
device rtwnfw
device rtwn_usb
device rtwn_pci
device wlan
device firmware
```

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

```
if_rtwn_pci_load="YES"
if_rtwn_usb_load="YES"
```

DESCRIPTION

The **rtwn** driver provides support for wireless network devices based on the Realtek RTL8192C, RTL8188E, RTL8192E, RTL8812A and RTL8821A programming APIs. These APIs are used by a wide variety of chips; most chips with USB and some with PCI interface are supported.

To enable use for PCI/PCIe systems, see the rtwn_pci(4) driver; for USB devices, use the rtwn_usb(4) driver.

The driver supports **station**, **adhoc**, **hostap** and **monitor** mode operation. There are no limitations for number of **monitor** mode virtual interfaces; in addition to any other virtual interface one **station** interface can be added (Note: RTL8821AU supports two non-monitor mode interfaces at the same time).

All chips have hardware support for WEP, AES-CCM and TKIP encryption.

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

FILES

/usr/share/doc/legal/realtek.LICENSE **rtwn** firmware license

The driver (if not compiled with **options RTWN_WITHOUT_UCODE**) may use following firmware files, which are loaded when an interface is brought up:

/boot/kernel/rtwn-rtl8188eefw.ko
/boot/kernel/rtwn-rtl8188eufw.ko
/boot/kernel/rtwn-rtl8192cfwE_B.ko
/boot/kernel/rtwn-rtl8192cfwE.ko
/boot/kernel/rtwn-rtl8192cfwT.ko
/boot/kernel/rtwn-rtl8192cfwU.ko
/boot/kernel/rtwn-rtl8192eufw.ko
/boot/kernel/rtwn-rtl8812aufw.ko
/boot/kernel/rtwn-rtl8821aufw.ko

EXAMPLES

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

```
ifconfig wlan create wlandev rtwn0 inet 192.168.0.20 \
    netmask 0xffffff00
```

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

```
ifconfig wlan create wlandev rtwn0 ssid my_net up
```

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

```
ifconfig wlan create wlandev rtwn0 ssid my_net \
        wepmode on wepkey 0x1234567890 weptxkey 1 up
```

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

```
ifconfig wlan create wlandev rtwn0 wlanmode adhoc ssid my_net \
            wepmode on wepkey 0x01020304050607080910111213 weptxkey 1 \
            channel 4
```

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

```
ifconfig wlan0 create wlandev rtwn0 wlanmode adhoc
ifconfig wlan0 inet 192.168.0.22 netmask 0xffffff00 ssid my_net \
            mode 11b
```

Create a host-based access point:

```
ifconfig wlan0 create wlandev rtwn0 wlanmode hostap
ifconfig wlan0 inet 192.168.0.10 netmask 0xffffff00 ssid my_ap
```

LOADER TUNABLES

Tunables can be set at the loader(8) prompt before booting the kernel or stored in loader.conf(5).

dev.rtwn.%d.hwcrypto

This tunable controls how key slots are assigned:
0 - disable h/w crypto support. Features that require access to frame contents (e.g., TCP/UDP/IP Rx checksum validation) will not work;
1 - use h/w crypto support for pairwise keys only;
2 - use h/w crypto support for all keys; may not work for multi-vap configurations.
By default it is set to 1.

dev.rtwn.%d.ratectl

This tunable switches between rate control implementations:
0 - no rate control;
1 - driver sends ’tx complete’ reports to net80211; algorithm is controlled via net80211;
2 - firmware-based rate control.
By default it is set to 1; however driver may choose another algorithm in case if it is not implemented
Currently selected algorithm is reported via *dev.rtwn.%d.ratectl_selected* read-only OID.

dev.rtwn.%d.rx_buf_size

(USB only) Controls size of temporary Rx buffer; smaller buffer size may increase number of interrupts.

DIAGNOSTICS

rtwn%d: could not read efuse byte at address 0x%x
rtwn%d: %s: cannot read rom, error %d There was an error while reading ROM; device attach will be aborted. This should not happen.

rtwn%d: failed loadfirmware of file %s For some reason, the driver was unable to read the microcode file from the filesystem. The file might be missing or corrupted. The driver will disable firmware-dependent features.

rtwn%d: wrong firmware size (%zu)
rtwn%d: %s: failed to upload firmware %s (error %d)
rtwn%d: timeout waiting for firmware readiness Firmware upload failed; the file might be corrupted. The driver will disable firmware-dependent features. This should not happen.

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

SEE ALSO

intro(4), netintro(4), rtwn_pci(4), rtwn_usb(4), rtwnfw(4), wlan(4), wlan_amrr(4), wlan_ccmp(4), wlan_tkip(4), wlan_wep(4), wlan_xauth(4), hostapd(8), ifconfig(8), wpa_supplicant(8)

HISTORY

The **urtwn** driver first appeared in OpenBSD 4.9 and FreeBSD 10.0; the **rtwn** driver first appeared in OpenBSD 5.8.

AUTHORS

The **rtwn** driver was initially written by Stefan Sperling <*stsp@openbsd.org*> and ported by Kevin Lo <*kevlo@freebsd.org*>. It was based on the **urtwn** driver written by Damien Bergamini <*damien.bergamini@free.fr*>.

BUGS

The **rtwn** driver currently does not implement firmware-based rate control.