

NAME

ieee80211_output - software 802.11 stack output functions

SYNOPSIS

```
#include <net80211/ieee80211_var.h>
```

int

```
M_WME_GETAC(struct mbuf *);
```

int

```
M_SEQNO_GET(struct mbuf *);
```

*struct ieee80211_key **

```
ieee80211_crypto_encap(struct ieee80211_node *, struct mbuf *);
```

void

```
ieee80211_process_callback(struct ieee80211_node *, struct mbuf *, int);
```

DESCRIPTION

The **net80211** layer that supports 802.11 device drivers handles most of the work required to transmit frames. Drivers usually receive fully-encapsulated 802.11 frames that have been classified and assigned a transmit priority; all that is left is to do crypto encapsulation, prepare any hardware-specific state, and push the packet out to the device. Outbound frames are either generated by the **net80211** layer (e.g. management frames) or are passed down from upper layers through the `ifnet(9)` transmit queue. Data frames passed down for transmit flow through **net80211** which handles aggregation, 802.11 encapsulation, and then dispatches the frames to the driver through its transmit queue.

There are two control paths by which frames reach a driver for transmit. Data packets are queued to the device's `if_snd` queue and the driver's `if_start` method is called. Other frames are passed down using the `ic_raw_xmit` method without queueing (unless done by the driver). The raw transmit path may include data frames from user applications that inject them through `bpf(4)` and NullData frames generated by **net80211** to probe for idle stations (when operating as an access point).

net80211 handles all state-related bookkeeping and management for the handling of data frames. Data frames are only transmit for a vap in the `IEEE80211_S_RUN` state; there is no need, for example, to check for frames sent down when CAC or CSA is active. Similarly, **net80211** handles activities such as background scanning and power save mode, frames will not be sent to a driver unless it is operating on the BSS channel with "full power".

All frames passed to a driver for transmit hold a reference to a node table entry in the `m_pkthdr.rcvif`

field. The node is associated with the frame destination. Typically it is the receiver's entry but in some situations it may be a placeholder entry or the "next hop station" (such as in a mesh network). In all cases the reference must be reclaimed with **ieee80211_free_node()** when the transmit work is completed. The rule to remember is: **net80211** passes responsibility for the *mbuf* and "node reference" to the driver with each frame it hands off for transmit.

PACKET CLASSIFICATION

All frames passed by **net80211** for transmit are assigned a priority based on any vlan tag assigned to the receiving station and/or any Diffserv setting in an IP or IPv6 header. If both vlan and Diffserv priority are present the higher of the two is used. If WME/WMM is being used then any ACM policy (in station mode) is also enforced. The resulting AC is attached to the mbuf and may be read back using the **M_WME_GETAC()** macro.

PAE/EAPOL frames are tagged with an M_EAPOL mbuf flag; drivers should transmit them with care, usually by using the transmit rate for management frames. Multicast/broadcast frames are marked with the M_MCAST mbuf flag. Frames coming out of a station's power save queue and that have more frames immediately following are marked with the M_MORE_DATA mbuf flag. Such frames will be queued consecutively in the driver's *if_snd* queue and drivers should preserve the ordering when passing them to the device.

FRAGMENTED FRAMES

The **net80211** layer will fragment data frames according to the setting of *iv_fragthreshold* if a driver marks the IEEE80211_C_TXFRAG capability. Fragmented frames are placed in the devices transmit queue with the fragments chained together with *m_nextpkt*. Each frame is marked with the M_FRAG mbuf flag, and the first and last are marked with M_FIRSTFRAG and M_LASTFRAG, respectively. Drivers are expected to process all fragments or none.

TRANSMIT CALLBACKS

Frames sent by **net80211** may be tagged with the M_TXCB mbuf flag to indicate a callback should be done when their transmission completes. The callback is done using **ieee80211_process_callback()** with the last parameter set to a non-zero value if an error occurred and zero otherwise. Note **net80211** understands that drivers may be incapable of determining status; a device may not report if an ACK frame is received and/or a device may queue transmit requests in its hardware and only report status on whether the frame was successfully queued.

SEE ALSO

bpf(4), ieee80211(9), ifnet(9)