NAME

ieee80211_beacon - 802.11 beacon support

SYNOPSIS

#include <net80211/ieee80211_var.h>

*struct mbuf **
ieee80211_beacon_alloc(*struct ieee80211_node *, struct ieee80211_beacon_offsets **);

int
ieee80211_beacon_update(*struct ieee80211_node *, struct ieee80211_beacon_offsets *, struct mbuf *, int mcast*);

void
ieee80211_beacon_notify(*struct ieee80211vap *, int what*);

DESCRIPTION

The **net80211** software layer provides a support framework for drivers that includes a template-based mechanism for dynamic update of beacon frames transmit in hostap, adhoc, and mesh operating modes. Drivers should use **ieee80211_beacon_alloc**() to create an initial beacon frame. The *ieee80211_beacon_offsets* structure holds information about the beacon contents that is used to optimize updates done with **ieee80211_beacon_update**().

Update calls should only be done when something changes that affects the contents of the beacon frame. When this happens the iv_update_beacon method is invoked and a driver-supplied routine must do the right thing. For devices that involve the host to transmit each beacon frame this work may be as simple as marking a bit in the *ieee80211_beacon_offsets* structure:

```
static void
ath_beacon_update(struct ieee80211vap *vap, int item)
{
	struct ieee80211_beacon_offsets *bo = &ATH_VAP(vap)->av_boff;
		setbit(bo->bo_flags, item);
}
```

with the **ieee80211_beacon_update**() call done before the next beacon is to be sent.

Devices that off-load beacon generation may instead choose to use this callback to push updates immediately to the device. Exactly how that is accomplished is unspecified. One possibility is to update the beacon frame contents and extract the appropriate information element, but other scenarios are

possible.

MULTI-VAP BEACON SCHEDULING

Drivers that support multiple vaps that can each beacon need to consider how to schedule beacon frames. There are two possibilities at the moment: *burst* all beacons at TBTT or *stagger beacons* over the beacon interval. Bursting beacon frames may result in aperiodic delivery that can affect power save operation of associated stations. Applying some jitter (e.g. by randomly ordering burst frames) may be sufficient to combat this and typically this is not an issue unless stations are using aggressive power save techniques such as U-APSD (sometimes employed by VoIP phones). Staggering frames requires more interrupts and device support that may not be available. Staggering beacon frames is usually superior to bursting frames, up to about eight vaps, at which point the overhead becomes significant and the channel becomes noticeably busy anyway.

SEE ALSO

ieee80211(9)