

## Re: Network drivers that don't suspend on interface down

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-12/msg06598.html>

---

- *From:* Dan Williams <[dcbw@xxxxxxxxxx](mailto:dcbw@xxxxxxxxxx)>
  - *Date:* Thu, 21 Dec 2006 12:16:06 -0500
- 

On Thu, 2006-12-21 at 14:19 +0100, Sven-Haegar Koch wrote:

On Wed, 20 Dec 2006, Dan Williams wrote:

If we define interface down as meaning that the device is powered down and the radio switched off, then (b) and (c) would presumably just need to ensure that the interface is downed. (a) is a slightly more special case – if the switch disables the radio, I guess we then want the driver to down the interface as well.

Correct.

In the (a) case, drivers should presumably refuse to bring the interface up if the radio is disabled?

Right; the driver simply can't do anything about it, because the switch is hardwired to the card and either the card's firmware takes care of it, or the chipset takes care of it. The driver has no say whatsoever in the state of the card's radio for this case. I tend to think this case is on it's way out in the same way that fullmac cards are falling out of favor (ie, do everything in software and save \$\$\$), but they are around and we need to support them.

In this case, down really does mean down too. The driver cannot honor requests to set SSID, frequency, etc, because it's simply not possible at that time.

What do you mean with this exactly?

## Re: Network drivers that don't suspend on interface down

Should the user not be able to set these values, or should the driver not be able to activate them?

I think it is correct when the driver does not activate them, but I think the user should be able to configure them, have them stored inside `cfg80211`/the driver, and have them activated/used when uping the interface, or when the `rkill` switch has been deactivated. Otherwise it will get impossible to boot with `rkill` disabled, toggle the switch later on and have everything working.

This would be an optimization. You could possibly `_set_` values, but obviously an `'associate'` command would fail, and so it should. But there's really not that much of a point to doing this, because `cfg80211` should support "packaging" up all the config for a particular association request into one call, and then just blasting that to the card. Ideally configuration wouldn't be pushed to the card piecemeal. As `WEXT` stands right now, setting the SSID on the card is essentially the `"associate"` command, which obviously wouldn't work when the card is down. `cfg80211` can fix that, you're right.

And another side to this:

if a disabled `rkill` switch downs the interface (opposed to just disabling it but staying `"ifconfig up"`) – what happens to the ip config of this interface? What reconfigures the needed routes when a re-enabled `rkill` switch reactivates the interface? Will manual route add and `ifconfig` statements be impossible and we'll get forced to use some crappy `distri`-scripts and daemons for it?

For anything other than unencrypted and WEP-only networks, you already need a userspace program to configure your wireless card. Dynamic WEP, LEAP, WPA, WPA2, 802.1x all require much, much more handshake and validation that should `_ever_` be in a driver. You should `_never_`, ever be configuring your wireless card with module parameters. I'm sure something like `iwconfig` would be fine to configure your card with.

When the card goes down, it normally loses its association to the access point anyway, and you need to start the association and authentication over completely. At that point, it's no longer guaranteed that you could ever get a previous IP address back.

What does downing an ethernet device do? It clears out routes associated with that device, and clears assigned addresses (I think?). Wireless is and should not be any different here. When you bring the device back up, you need to go through some amount of renegotiation anyway.

Re: Network drivers that don't suspend on interface down

And third point just coming to my mind:

how is changing the mac address of the card supposed to work? Changing it through ifconfig only works when the interface is downed, so the newly wanted mac address has to be saved somewhere before the interface is reenabled and reinitialized on the next "ifconfig up".

(And I think it is an absolute requirement that NO packet with the old/default mac address may be sent into the air whatsoever)

That's how it should work. If you want to change the MAC address, the card shouldn't probably be down.

Dan

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>