New ifnet(9) KPI

Gleb Smirnoff
glebius@FreeBSD.org

BSDCan 2015
Ottawa

June 11, 2015

project wiki page and code

https://wiki.freebsd.org/projects/ifnet
svn+ssh://svn.freebsd.org/base/projects/ifnet

struct ifnet

- Glue between a driver and the stack
- Used to be embedded into driver softc, now softc and ifnet has pointers to each other

struct ifnet

- Glue between a driver and the stack
- Used to be embedded into driver softc, now softc and ifnet has pointers to each other
- At attach time driver and stack fill in the structure
- At run time driver changes different flags

Why changes are needed

- Extension of the struct ifnet could require recompilation of all drivers
- Editing struct ifnet could require patching all drivers

The opaque ifnet

Code snippet 1: net/if.h:

```
typedef struct ifnet * if_t;
```

Code snippet 2: net/if_var.h:

```
struct ifnet {
  /* actual structure definition */
};
```

The opaque ifnet

- Easy way: substitute any access to struct ifnet fields with a function

The opaque ifnet

- Easy way: substitute any access to struct ifnet fields with a function
- Hard way: design new KPI

what's in struct ifnet?

	driver access	instance specific
if_flags, if_capenable, if_mtu	R	y
link state, baudrate	RW	y
counters	W	y
if_capabilities, if_tso*	W once	y/n
address lists	R	y
driver name, if_clone	W once	n
methods	W once	n
if_type, dlt_type, header len	W once	n
if_media	W once *	y/n

what's else in struct ifnet?

The stuff the driver shouldn't be interested at

- Layer 2 softc: if_l2com
- Address family softcs: if_afdata[AF_MAX]
- Bunch of other softcs: if_lagg, if_carp, if_netmap, etc

what's else in struct ifnet?

The stuff the driver shouldn't be interested at

- Layer 2 softc: if_l2com
- Address family softcs: if_afdata[AF_MAX]
- Bunch of other softcs: if_lagg, if_carp, if_netmap, etc

This all can be generalized!

new if_attach()

~~if_alloc()~~ if_attach(if_attach_args *args)

- sleeps, doesn't fail (save name conflict)
- if_attach_args is versioned
- if_attach_args contains all the "W once" stuff
  - ifdriver pointer
  - softc pointer
  - lladdr
  - supported media list
  - capabilities, TSO limits
  - initial values: MTU, capenable, baudrate, etc

ifdriver, ifops

struct ifdriver is static in the driver, storing all non-instance specific stuff

- Interface methods in special structure ifops
- Name, type, DLT, headerlength

ifmedia

Code snippet 3: net/if_media.h:

```
typedef int if_media_t;
```

- Drivers declare static/dynamic array of if_media_t, pointed to from if_attach_args
- Drivers declare if_media_change_t, if_media_status_t in ifops
- Implementation opaque and private to if_media.c

ifmedia + miibus

- miibus(4) is completely ifnet(9) agnostic
- mii_attach() allocates pointer to if_media_t array, to be used later in if_attach_args

- if_init is no longer a method, becomes static function
- if_poll for polling(4)
- ~~if_start~~
- if_transmit doesn't m_freem() in case of error

if_flags, if_capenable, if_mtu (R, instance specific)

- if_ioctl method is the only channel to modify the values
- stack does sanity checking
- driver may refuse new value
- if driver accepts value, it may cache it

~~if_drv_flags~~

- IFF_DRV_OACTIVE goes away together with if_start and generic queue
- IFF_DRV_RUNNING goes to driver softc, protected by driver lock

link state, baudrate

```
void    if_setbaudrate(if_t, uint64_t);
void    if_link_state_change(if_t, int);
```

counters

- if_inc_counter() is already in head
- Driver is 100% resposible for the counters
- TX counters are updated on TX completion, not on enqueue
- Use if_inc_txcounters() for TX update

traversing address lists

```
typedef void
ifaddr_cb_t(void *, struct sockaddr *,
    struct sockaddr *, struct sockaddr *);
typedef void
ifmaddr_cb_t(void *, struct sockaddr *);

void if_foreach_addr(if_t, ifaddr_cb_t,
    void *);
void if_foreach_maddr(if_t, ifmaddr_cb_t,
    void *);
```

net80211 drivers

- No ifnet layer in the driver
- Task is 50% done
- Will be committed to head separately

`https://wiki.freebsd.org/projects/ifnet/net80211`

drivers that do if_start

That's 80-90% of all legacy drivers

- if_start and struct ifqueue go away
- driver opts-in for a generic queue in if_attach
- stack provides: if_snd_len(), if_snd_enqueue(), if_snd_dequeue() and if_snd_prepend()
- driver's if_transmit is a short copy&paste ☹

lagg(4)

- now lagg(4) hijacks if_transmit of an interface
- it will hijack ifops
- ifops can be stacked, like VOPs

ALTQ

- now ALTQ works on top of ifqueue
- it will hijack ifops

scope of work

- There are > 200 drivers to be converted
- Only 16 has been converted so far
- Conversion of typical 100Mbit driver takes 1 hour
- Usually driver is reduced by 100 LOC

`https://wiki.freebsd.org/projects/ifnet/progress`

open tasks

- Better generic queueing/transmit code?
- ALTQ
- lagg(4)
- VIMAGE

Your feedback & help is needed!

- Reviewing
- Criticizing, proposing better KPIs
- Converting drivers
- Testing