FreeBSD ifnet API

FreeBSD Developer Summit
Ottawa, Canada
May 16, 2007

What is it?

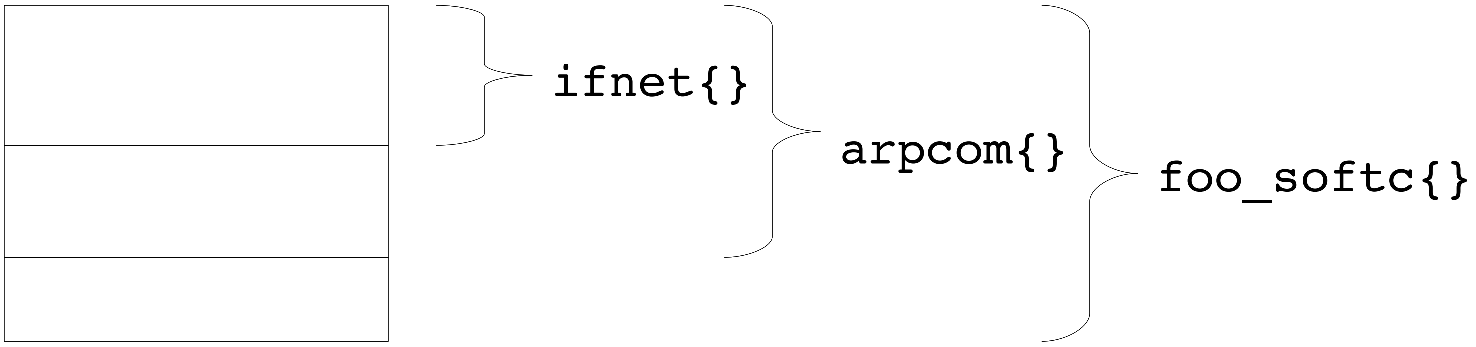
- Generic network interface code
  - Creation and registration
    - if_alloc(), if_attach()
  - Destruction
    - if_detach(), if_free()
  - Common data storage
  - OO interface to driver
    - Send/receive packets
    - Manipulate interface
- See also: *TCP/IP Illustrated, Volume 2*
  - Chapter 3, (4, 5)

Code

- The core interface
  - `sys/net/if.[ch]`
  - `sys/net/if_var.h`
- Pseudo-device Cloning
  - `sys/net/if_clone.[ch]`
- Layer 2 Common Code
  - Ethernet
    - `sys/net/ethernet.h`
    - `sys/net/if_arp.h`
    - `sys/net/if_ethersubr.c`

De-nesting of `struct ifnet`

Classic

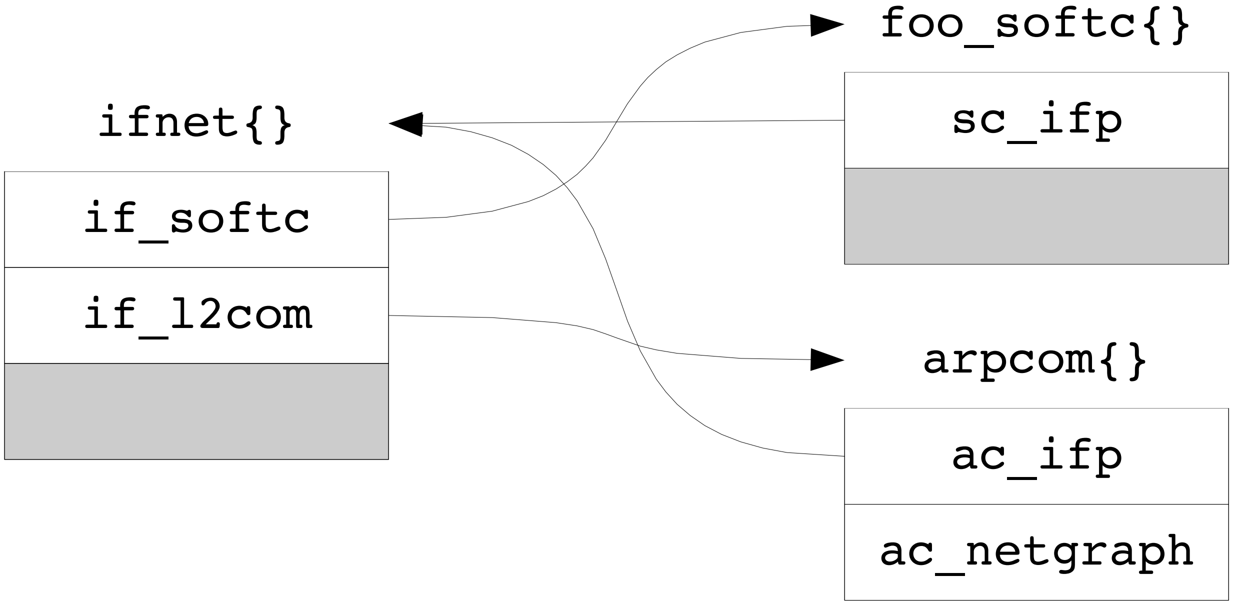


ifnet{}

arpcom{}

foo_softc{}

Modern



ifnet{}

if_softc

if_l2com

foo_softc{}

sc_ifp

arpcom{}

ac_ifp

ac_netgraph

Interface life cycle

- Creation
  - Hardware is probed or a clone is requested
  - `f_alloc(ITF_`*`XXX`*`)`
    - allocate `struct ifnet` instance
  - `if_attach(ifp)`, `ether_ifattach(ifp, MAC)`, etc
    - Attach the interface and the L2 code
- *Interface configured, packets sent/received, etc*

Interface life cycle (cont)

- Destruction
  - Hardware removed or the destruction of a clone is requested
  - `if_detach(ifp), ether_ifdetach(ifp),` etc
    - Detach the interface
  - `if_free(ifp), if_free_type(ifp)`
    - Free the `struct ifnet` and layer 2 common structure
    - `if_free_type()` should probably should be replaced with a field in the `struct ifnet`

Issues

- Race between `if_grow()` and `*_byindex()`
- Ordering of `if_detach()`
  - hooks to everywhere make things complicated
  - should we make the `ifnet if_dead`, defer `if_free()` or what?
- `if_init()` takes `void*` pointer to softc