Yandex

Yandex

Improving FreeBSD packet forwarding

Alexander Chernikov
Network engineer

Current L2/L3 problems

- Forwarding is slow
  - No multipath
  - FIBs still not adopted
- Netgraph does not scale
- No new “advanced” features
  - Except netmap :)

High-level overview

Netgraph

- Great idea, but non-scalable implementation
- Topology protection
  - ng_address_hook() → ng_topo_mtx
  - refcount(9) for both hooks/nodes
- Ideas
  - Convert ng_topo_mtx to rwlock?
  - Use counter(9) instead of refcounts?

Netisr

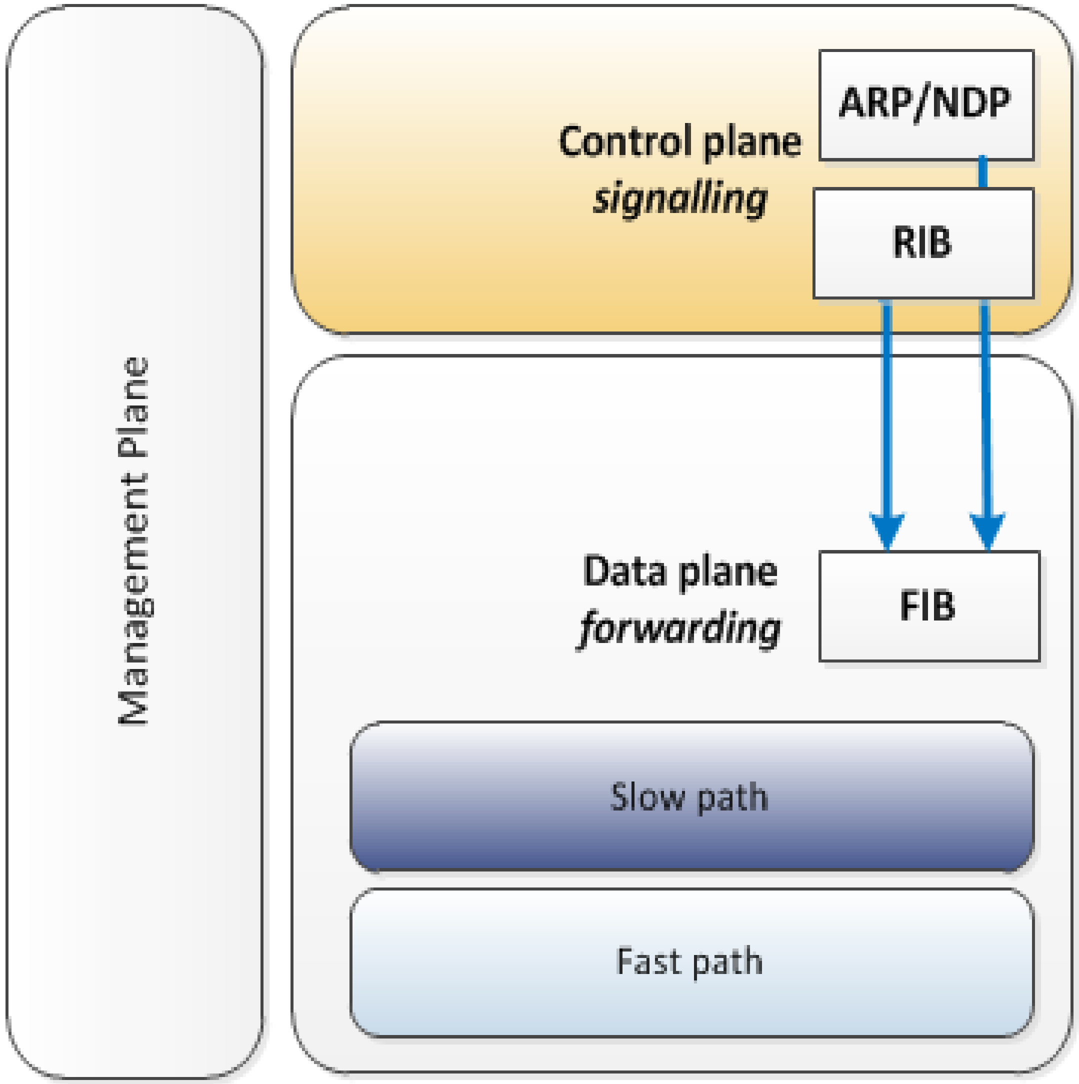
- Great idea
  - Modular way to deal with different packet types
  - Ability to do fine-grained flowid generation
- Performance
  - Per-packet queue lock/unlock
- Ideas
  - Add flowid generators
  - Use batches?

Counters(9)

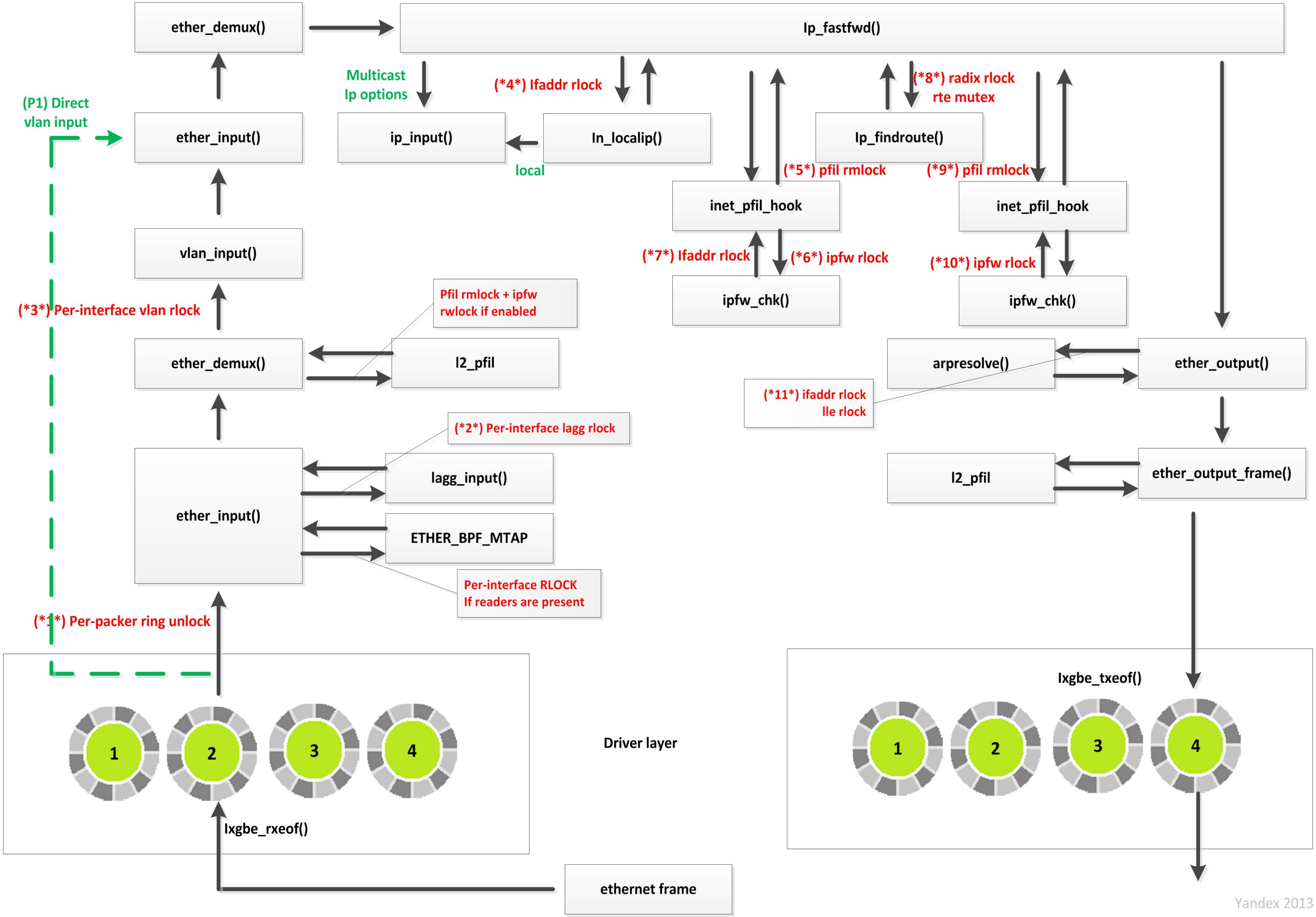
- += problem
  - Cache line contention
  - Losing accuracy: 4mpps counted instead of 5 real
- Current counters(9) consumers
  - struct ipstat / ip6stat
  - if_lagg(4)
- Next?
  - Drivers should care about stats, not stack
  - Use HW counters if possible
  - Use counters(9) as default counting mechanism
  - Remove ether_input_internal() ifp->if_ibytes += m->m_pkthdr.len;

Forwarding

Control / Data plane



- Separate control/data flows
- Different route DBs
- Optimize fast path
  - ip_fastforward()
  - Route(9) / rtenrty(9)



ether_demux()

Ip_fastfwd()

Multicast
Ip options

(*4*) Ifaddr rlock

(*8*) radix rlock
rte mutex

(P1) Direct
vlan input

ether_input()

ip_input()

In_localip()

Ip_findroute()

local

(*5*) pfil rmlock

(*9*) pfil rmlock

inet_pfil_hook

inet_pfil_hook

vlan_input()

(*7*) Ifaddr rlock

(*6*) ipfw rlock

(*10*) ipfw rlock

Pfil rmlock + ipfw
rwlock if enabled

ipfw_chk()

ipfw_chk()

(*3*) Per-interface vlan rlock

ether_demux()

l2_pfil

arpresolve()

ether_output()

(*11*) ifaddr rlock
lle rlock

(*2*) Per-interface lagg rlock

lagg_input()

l2_pfil

ether_output_frame()

ether_input()

ETHER_BPF_MTAP

Per-interface RLOCK
If readers are present

(*1*) Per-packer ring unlock

Ixgbe_txeof()

1 2 3 4

Driver layer

1 2 3 4

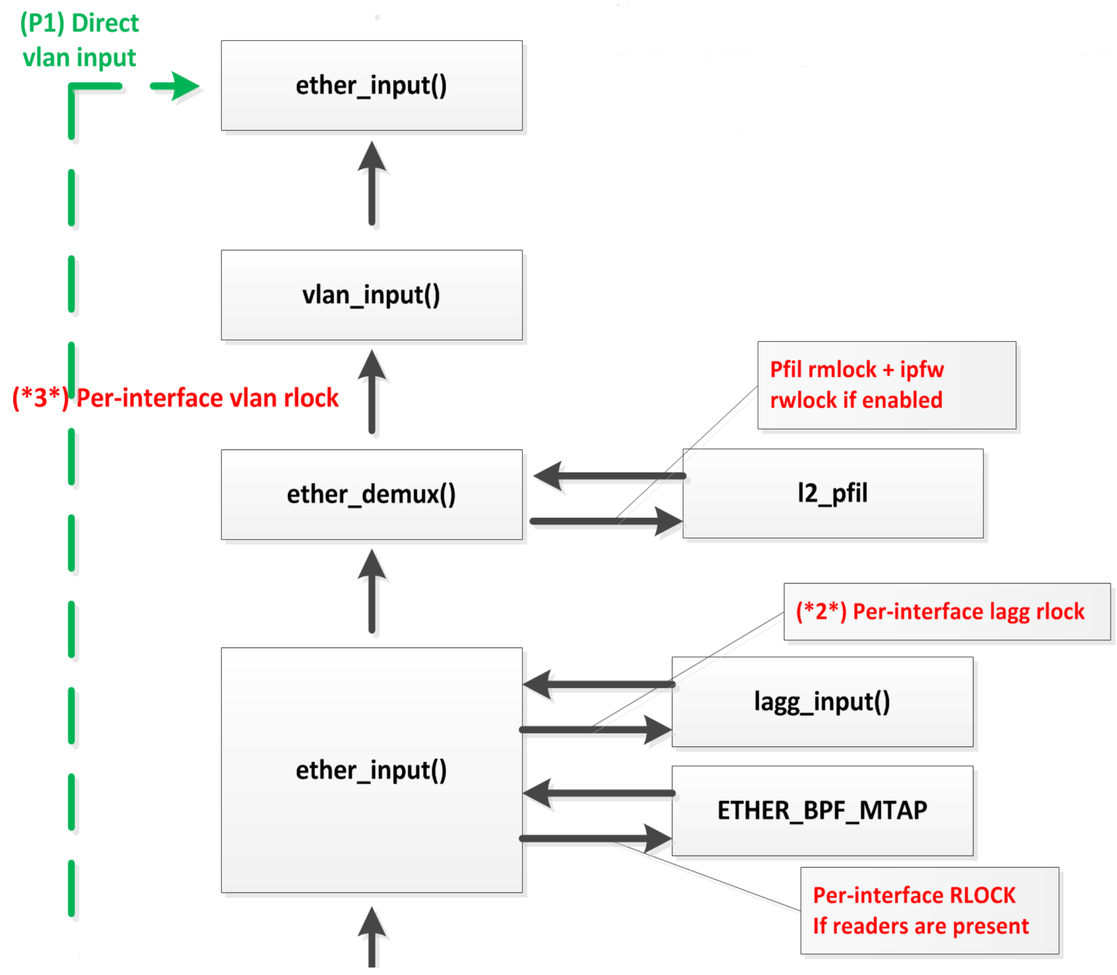
Ixgbe_rxeof()

ethernet frame

Yandex 2013

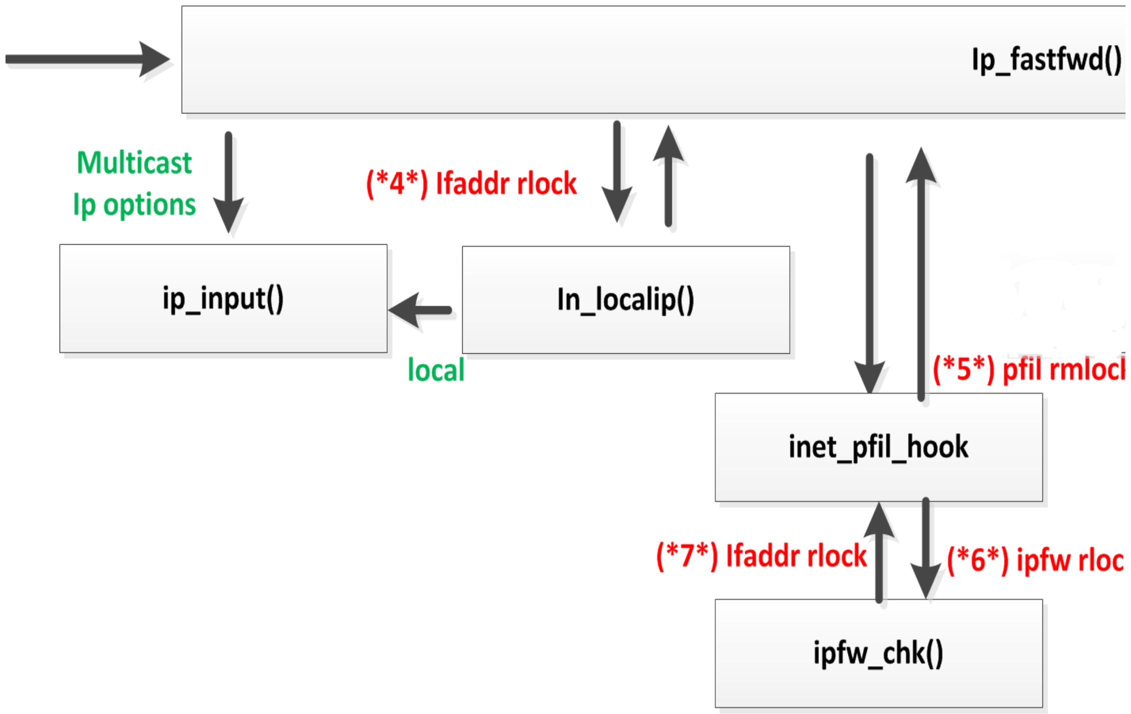
IPv4 forwarding scheme

Packet flow: L2 ingress


(P1) Direct vlan input
ether_input()
vlan_input()
(*3*) Per-interface vlan rlock
Pfil rmlock + ipfw rwlock if enabled
ether_demux()
l2_pfil
(*2*) Per-interface lagg rlock
lagg_input()
ether_input()
ETHER_BPF_MTAP
Per-interface RLOCK If readers are present

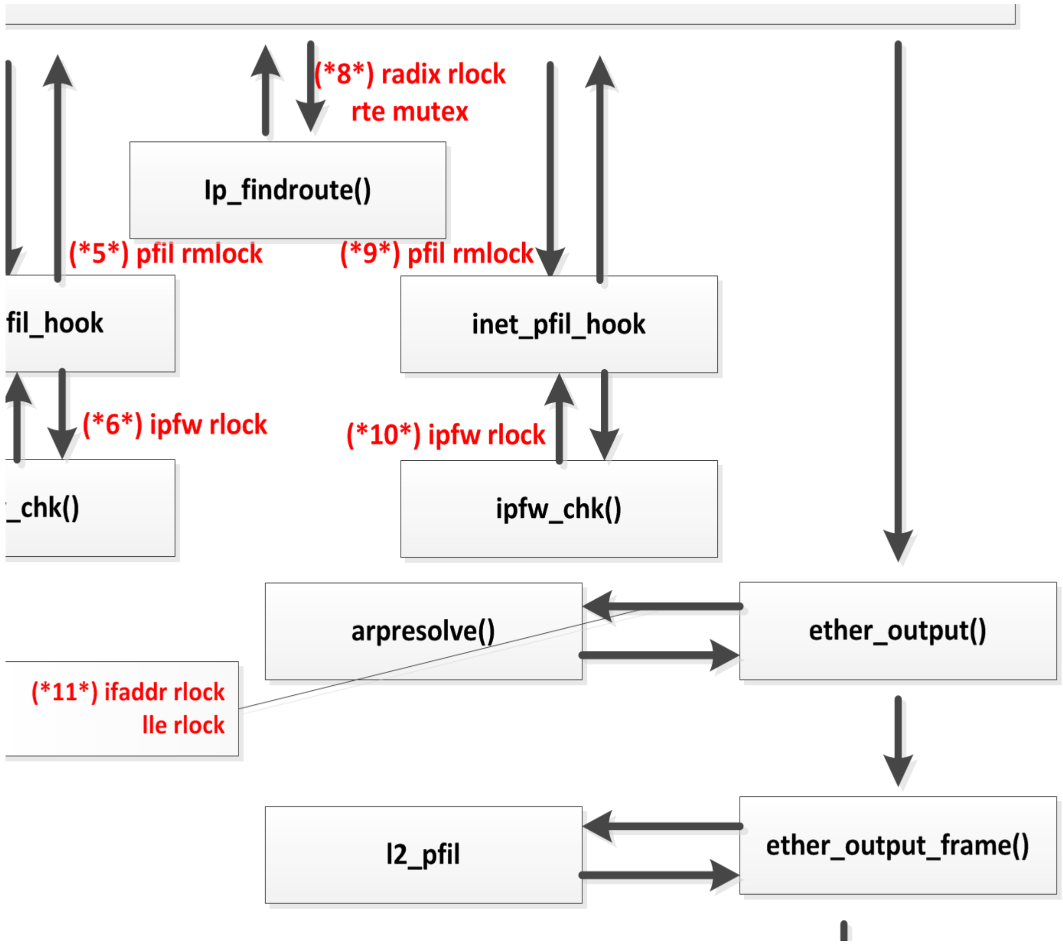
- Virtual interfaces
  - Should exist as control plane objects
- Lagg(4)
  - No lagg-specific errors on data path
  - No need to push ingress traffic thru
- Vlan(4)
  - No need to push ingress traffic thru
  - Use improved VLANHWFILTER

Packet flow: L3 ingress



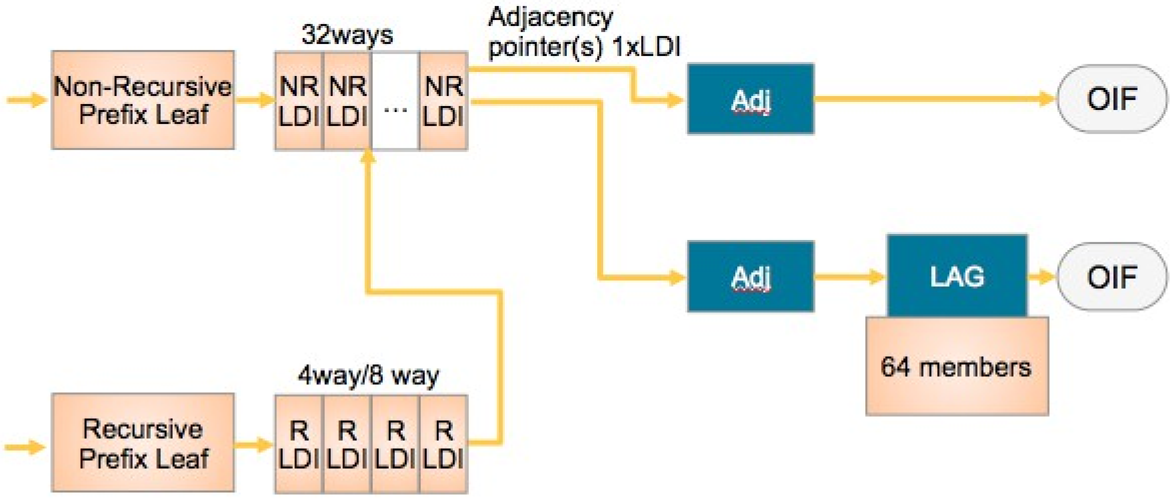
- Local packet?
  - Make check faster (lockless?)
  - Save result to mbuf?
- Utilize PFIL lock
  - Ipfw can use PFIL as main lock

Packet flow: L3 egress



- Eliminate rte lock
- Cover lladdr by radix lock
- Protect from if destroy

L3 egress: vendors



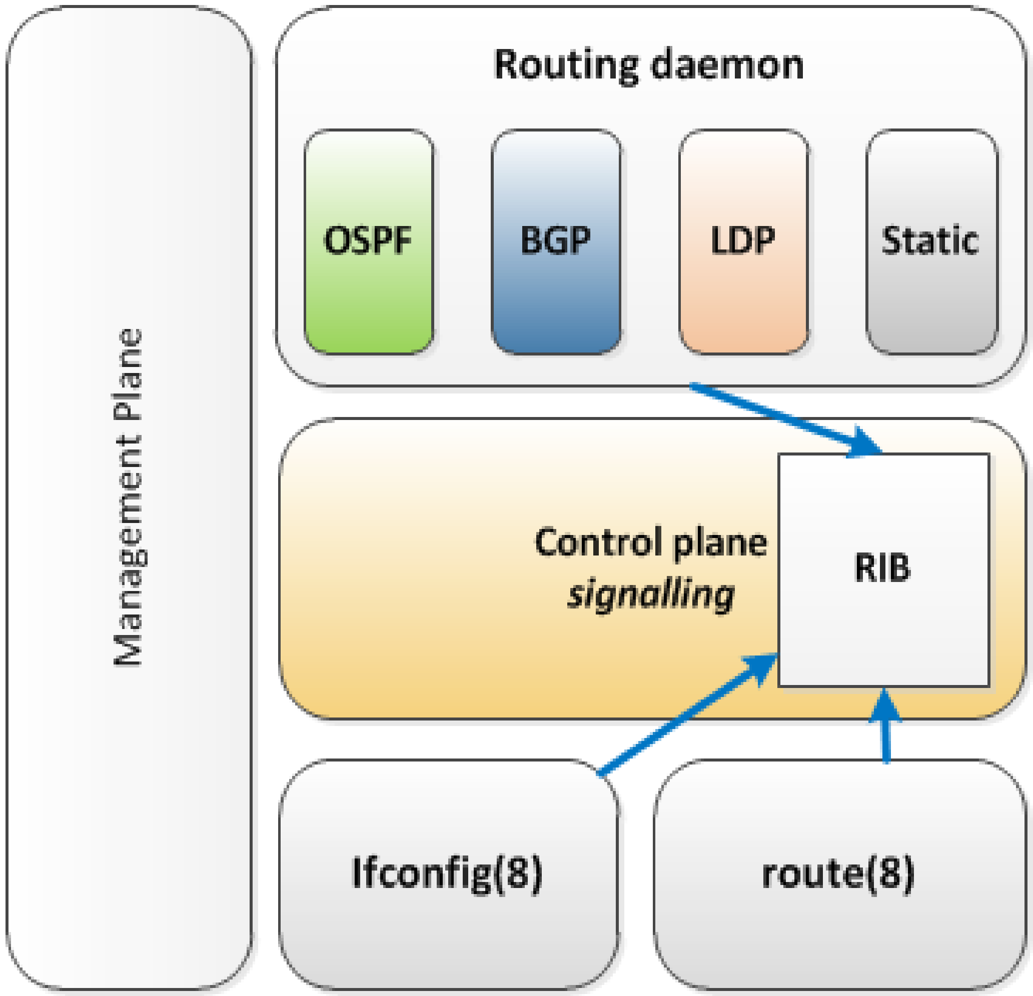
- IPv4 CEF (Cisco)
  - 8-8-8-8 multibit trie for nhops
  - Host cache (connected routes)
  - Adj structures for nhop info
  - L3 multipath via adj group
  - L2 multipath inside adj
- IPv4 fwd (Juniper contrail)
  - Same 8-8-8-8
  - Abstract 'NEXTHOP' structure
  - Every rewrite/encap stored there

RIB/FIB

Current problems

- Too slow/abstract to be FIB
- Not enough features (TCP*, ..)
- Rte locking is abused
- L2 rewrite requires 2 additional locks
- Egress interface removal problem

Kernel RIB features



- ECMP
- Extendable rte attrs
  - TCP MSS?
  - Process PID?
  - Source Ipv4/IPv6?
- Stack encap
  - MPLS
  - VxVLAN/NvGRE?
  - Qinq?

Kernel FIB features

- AF-dependent
  - IPv6: No link-local addresses in FIB
  - IPv6: Lookup on upper 64 bits
  - IPv4: Use 8-8-8-8 scheme or DXR or..
  - MPLS: lookup by array index
- Unified API
  - Any scheme can be loaded as a module
  - Integrated with L3->L2 mappings

Route socket

- Not easy extendable
  - Last change: 2009
  - No ABI keeping scheme
- No FIB/MRT support
- Ideas?
  - Add TLV support?
  - Or implement RFC 3549 (netlink)?
  - libmnl: small LGPL 2.1 implementation

Future

SDN/NFV train

- A lot of hype about SDN
- We still can get here
  - Firewalling element/function
  - Host VM “PE” router
  - Switch/Router operating system
- We need most of the above to get there

Yandex

Alexander Chernikov

Network engineer

melifaro@yandex-team.ru

melifaro@FreeBSD.org

Спасибо

References

[1] Cisco ASR9000 loadbalancing architecture: https://supportforums.cisco.com/docs/DOC-26687

[2] Cisco CEF: http://www.cisco.com/en/US/tech/tk827/tk831/technologies_white_paper09186a00800a62d9.shtml#express

[3] BGP Prefix Independent Convergence: http://www.ietf.org/proceedings/85/slides/slides-85-rtgwg-10

[4] Juniper Contrail vRouter: https://github.com/Juniper/contrail-vrouter/

[5] L3VPN on end-system: http://tools.ietf.org/html/draft-ietf-l3vpn-end-system-01

[6] Cumulus Switch OS: http://cumulusnetworks.com/product/overview/

[7] Minimalist Netlink library: http://netfilter.org/projects/libmnl/

[8] Intel DPDK: http://www.intel.com/content/dam/www/public/us/en/documents/presentation/dpdk-packet-processing-ia-overview-presentation.pdf

[9] DXR route lookup scheme: http://info.iet.unipi.it/~luigi/papers/20120601-dxr.pdf