

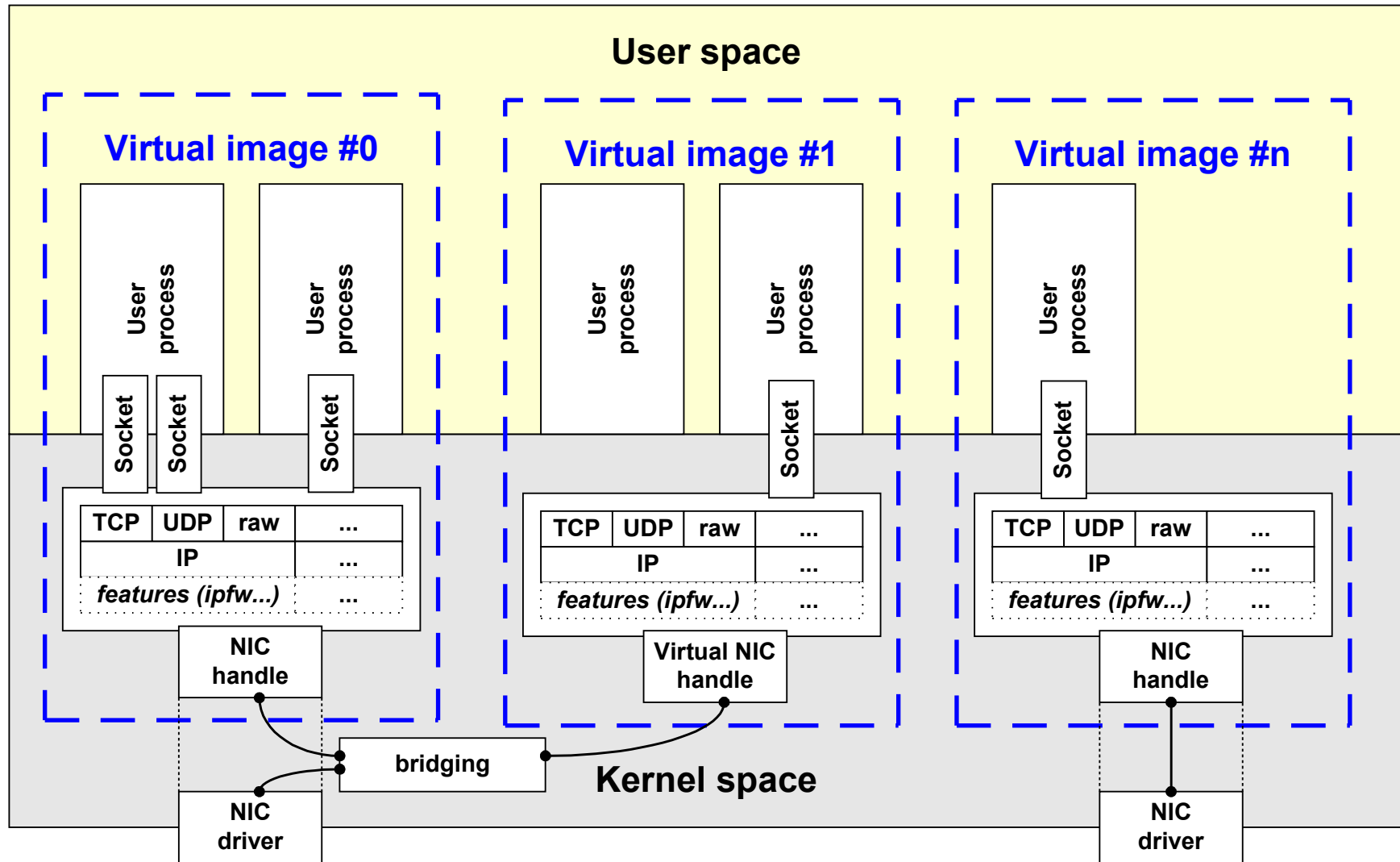
Network stack virtualization

Marko Zec

zec@fer.hr

University of Zagreb

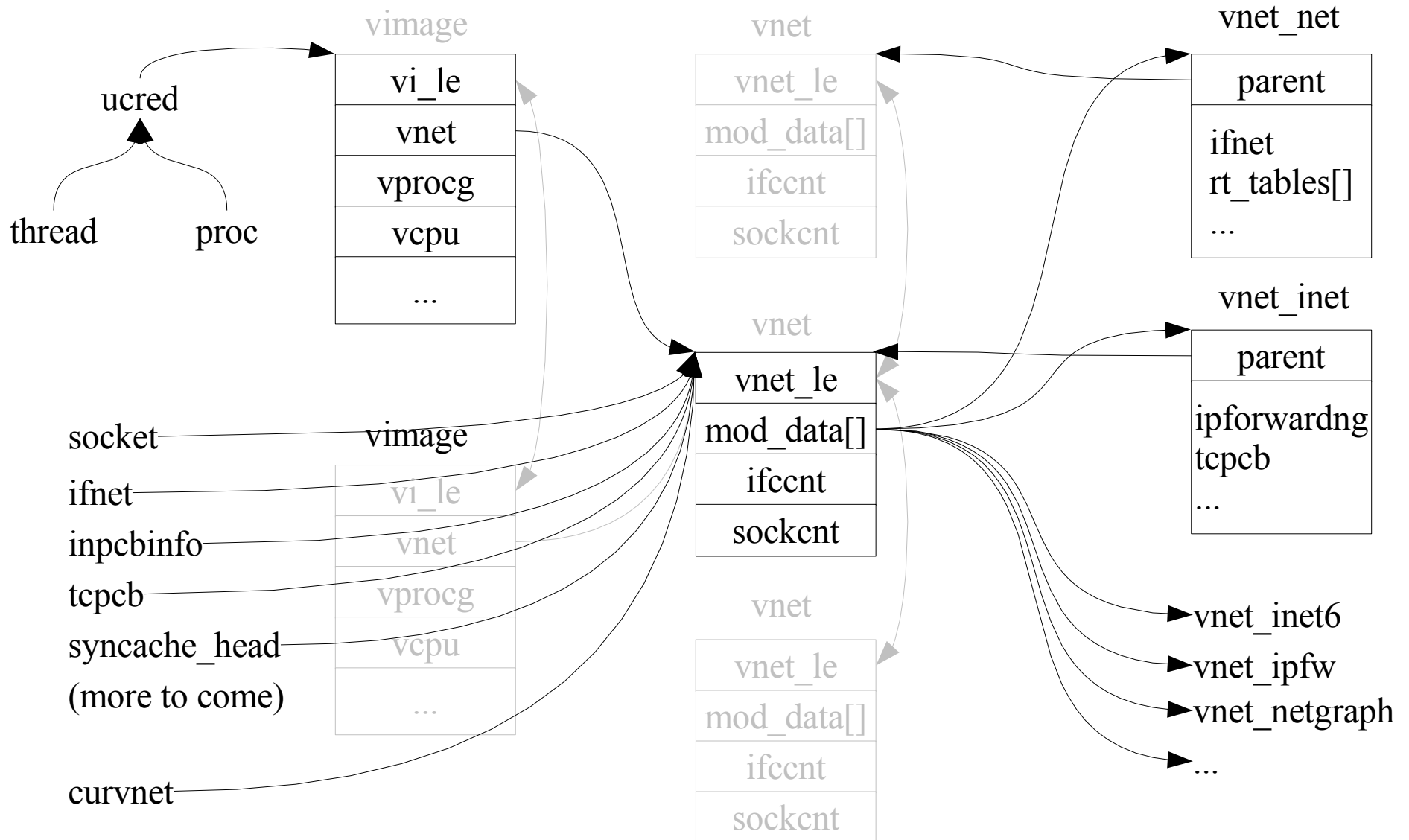
The basic idea: replicate global networking state



(Re)implementation: 7.0-CURRENT

- Goals:
- Conditional compilation
- Better support for kernel loadable modules
- Scope of changes is huge: reduce code churn
- SMP must work
- Otherwise, no chances for including the changes into main FreeBSD tree
- Target: was 7.0-RELEASE, 8.0 more realistic?

Replicate global networking state: how?



Conditional compilation: option VIMAGE

- Dereference virtualized symbols: how?
 - Use macros for this. Example:
 - `if_addrhead` becomes `v_if_addrhead`
 - Standard kernel:
 - `V_if_addrhead` expands back to `if_addrhead`
 - Virtualized kernel:
 - `V_if_addrhead` expands to `vnet_net->_if_addrhead`
 - `Sysctl` and `kldsym` interfaces extended to support access to virtualized symbols

Reducing code churn

- Implicitly pass the `vnet` context to operate on:
 - Thread-local `curvnet` variable

```
void if_attach(struct ifnet *ifp)
{
    INIT_VNET_NET(curvnet);
    ...
}
```

`INIT_VNET_NET(x)` (`x` is a `struct vnet *`) expands to

```
struct vnet_net *vnet_net = x->mod_data[VNET_MOD_NET];
```

Generalizing OS-level virtualization

- Management concepts / API
 - Top-level resource container `struct vimage`
 - Contains freely combinable subsystem-specific state
 - `vnet`, `vcpu`, `vprocg`...
 - Single process with sockets in multiple stacks
 - Extend socket interface -> multi-table routing daemons
 - Hierarchy of vimages – follow UNIX process model?
 - Permissions, restrictions, inheritance...
 - How to best integrate those new concepts / features with the rest of the system?

Project status

- Supported by NLNet and FreeBSD foundation
 - Started in August 2006, should have already finished...
- In sync with -CURRENT: p4 projects/vimage
 - Snap-in replacement kernel – no userspace changes!
 - <http://imunes.tel.fer.hr/virtnet/> : CVSup
- Reasonably stable already
 - Lots to be done: locking, management API & housekeeping
- Most important networking subsystems virtualized:
 - IPv4, IPv6, NFS, IPFW / PF firewalls, BPF, raw / routing sockets...
- Long-term fate?