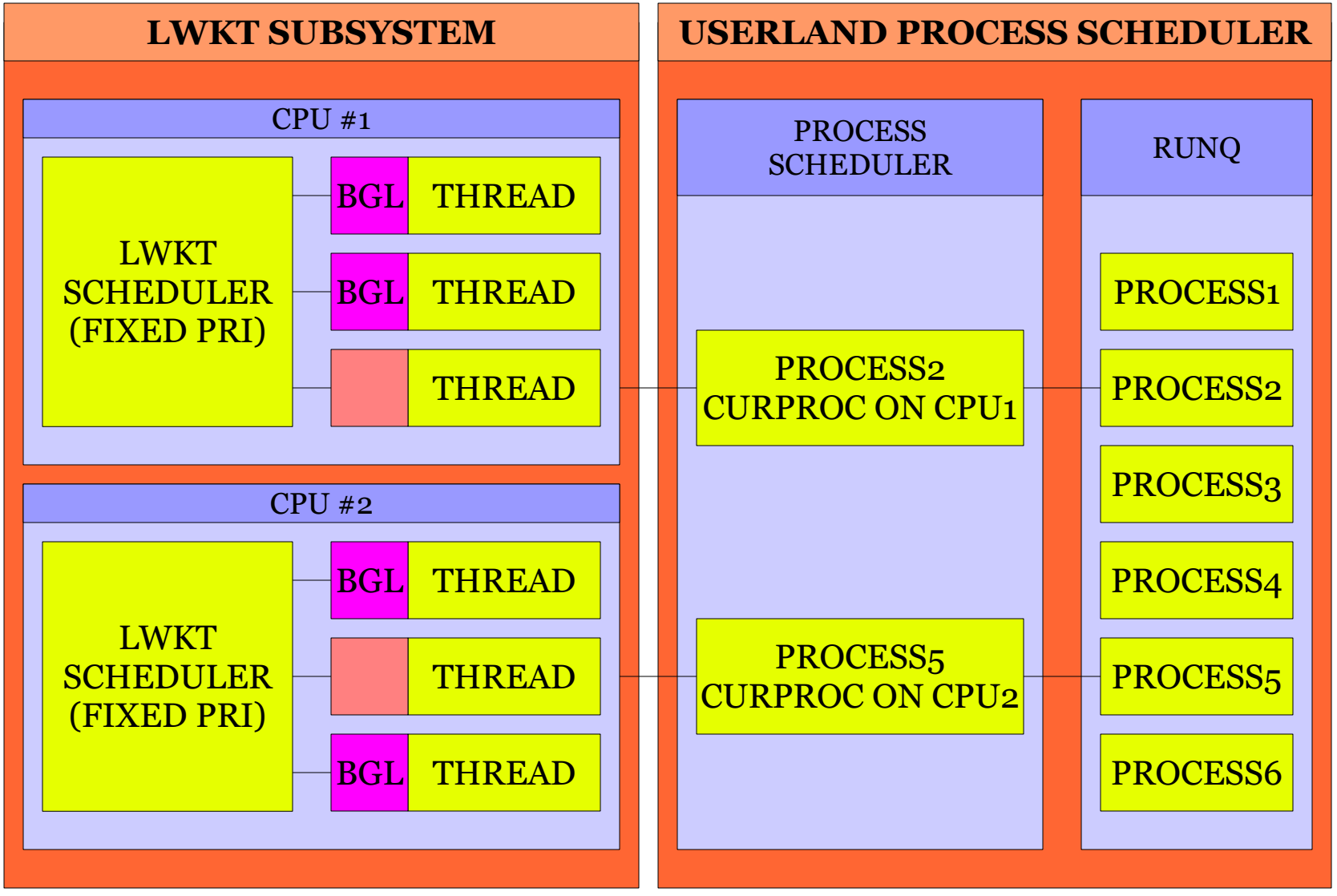
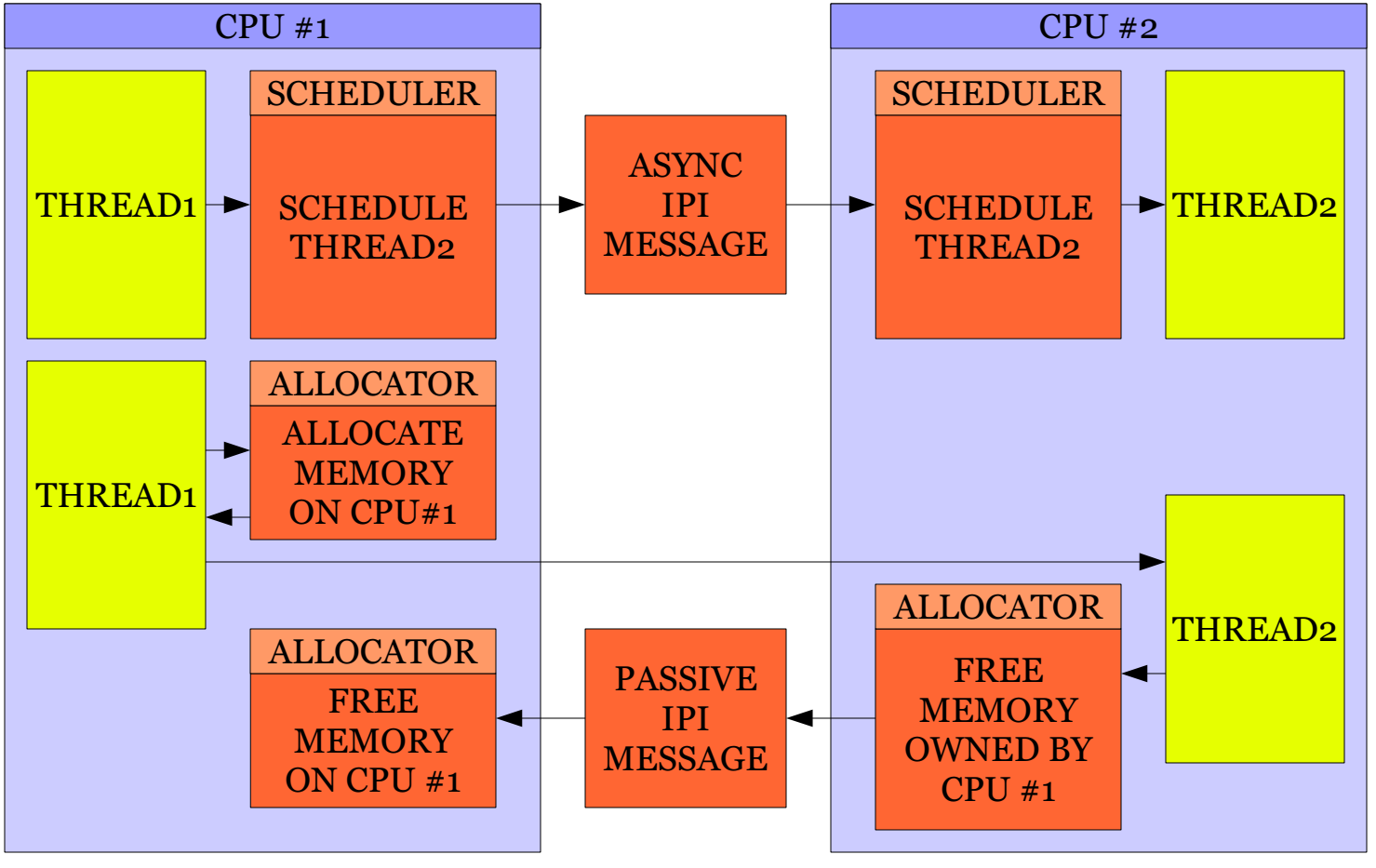

WWW.DRAGONFLYBSD.ORG

# Light Weight Kernel Threading and User Processes



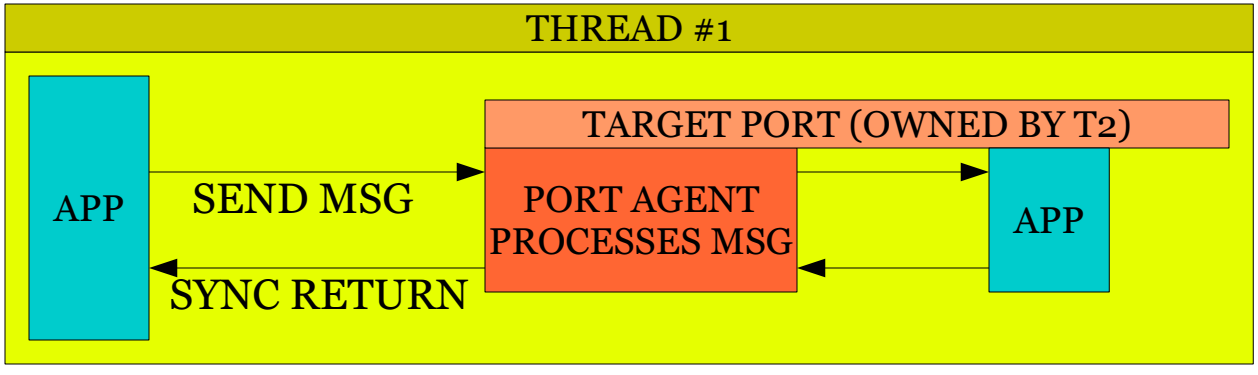
# IPI Messaging

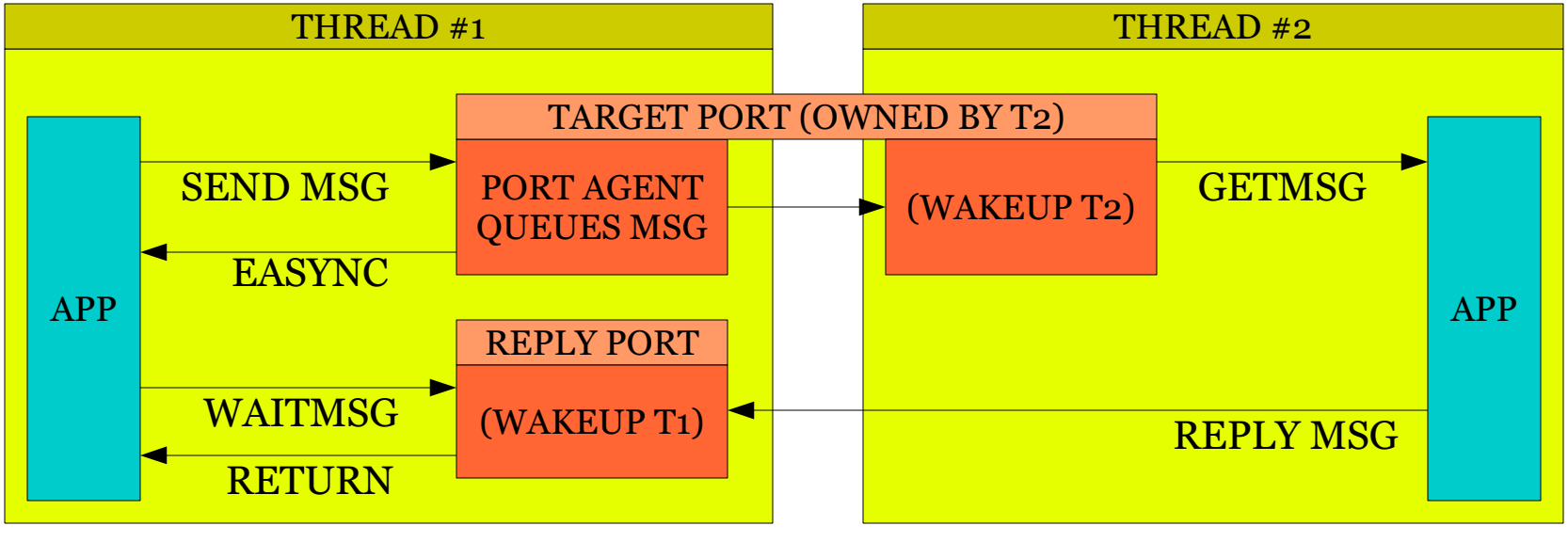
- Abstraction promotes cpu isolative algorithms.
- IPI Messages avoid mutexes. Software crossbar approach.
- Critical sections to interlock against interrupts and IPIs.
- Many IPIs can be passive. Pipelining can be made optimal.



# Light Weight Kernel Thread Messaging

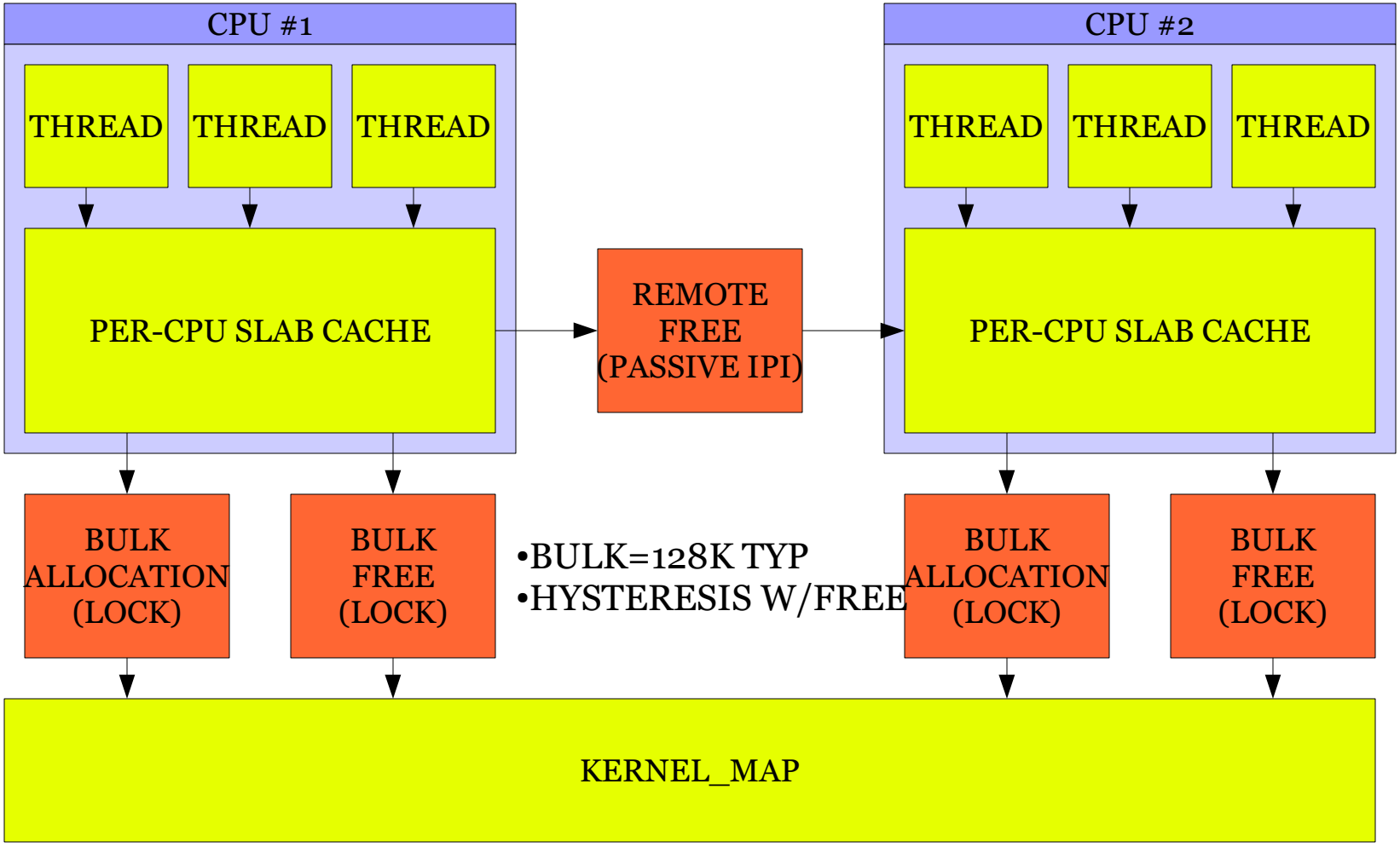
- Amiga style message and port API.
- Semisynchronous: direct call to target's port agent.
- Very fast synchronous path.
- Highly Flexible.
- UP/SMP optimized



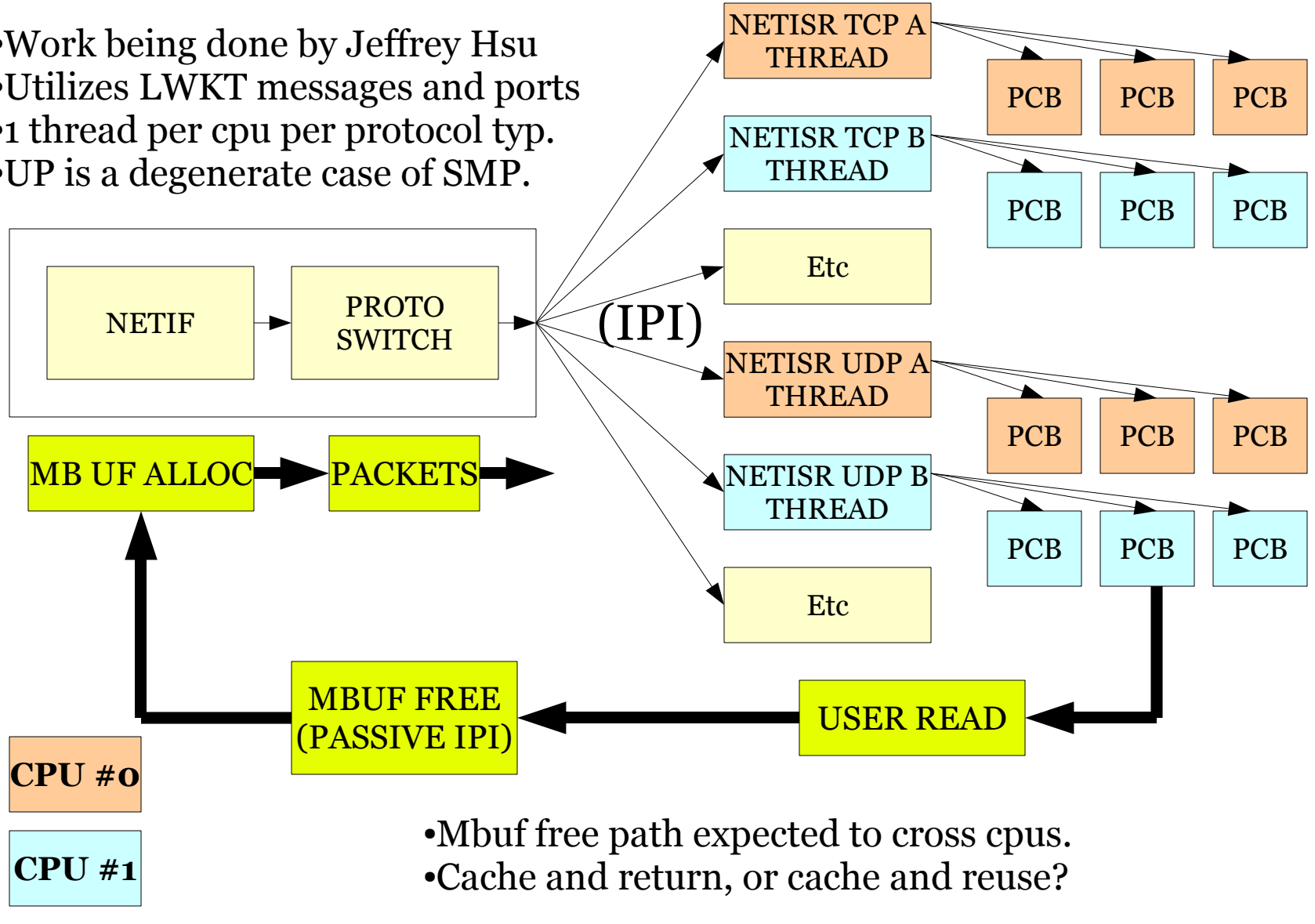
# Slab Allocator

- Per-CPU localization.
- Backed by kernel_map.
- No more kmem_map.



# Threaded Network Protocol Stacks

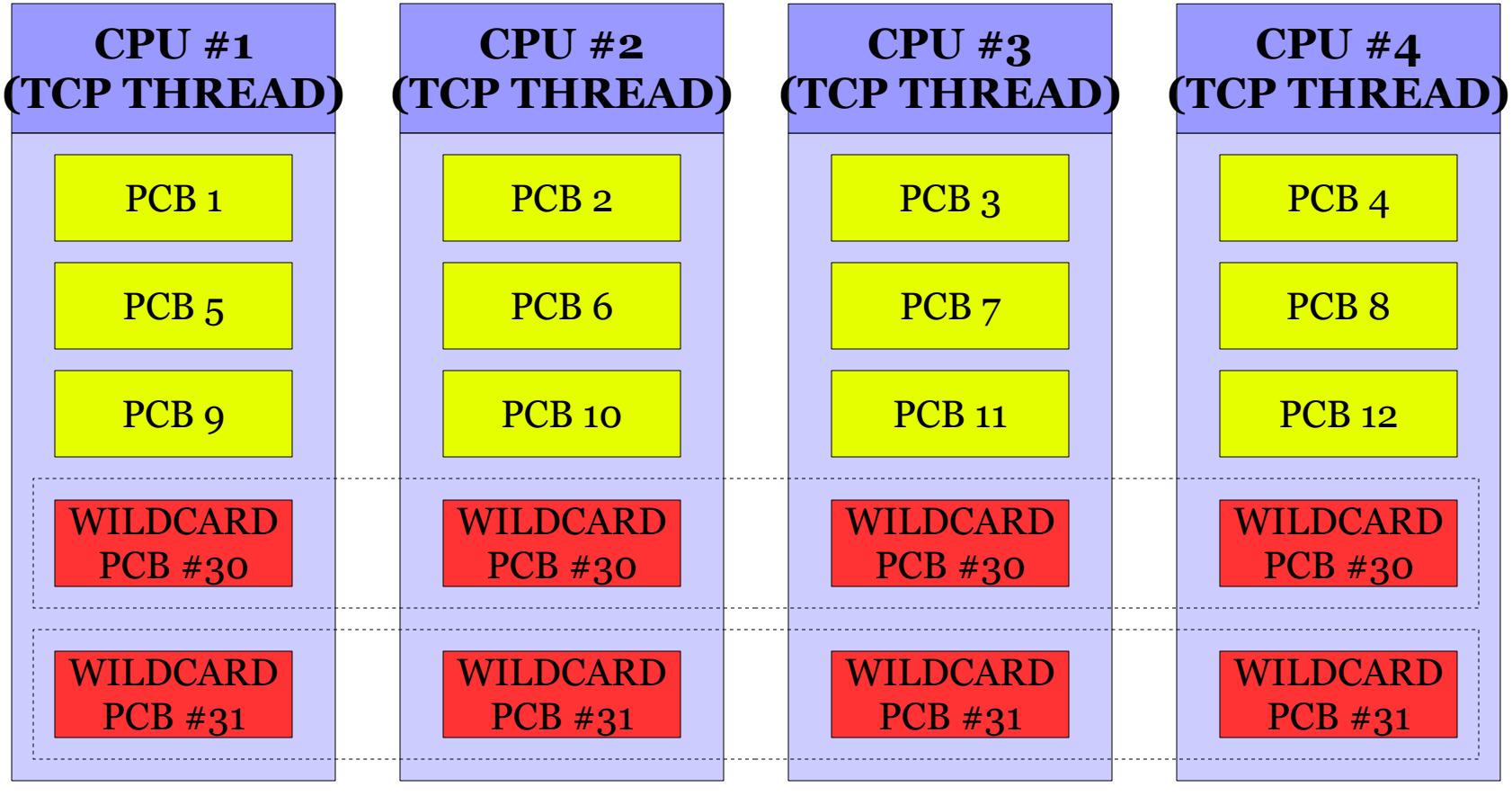
- Work being done by Jeffrey Hsu
- Utilizes LWKT messages and ports
- 1 thread per cpu per protocol typ.
- UP is a degenerate case of SMP.



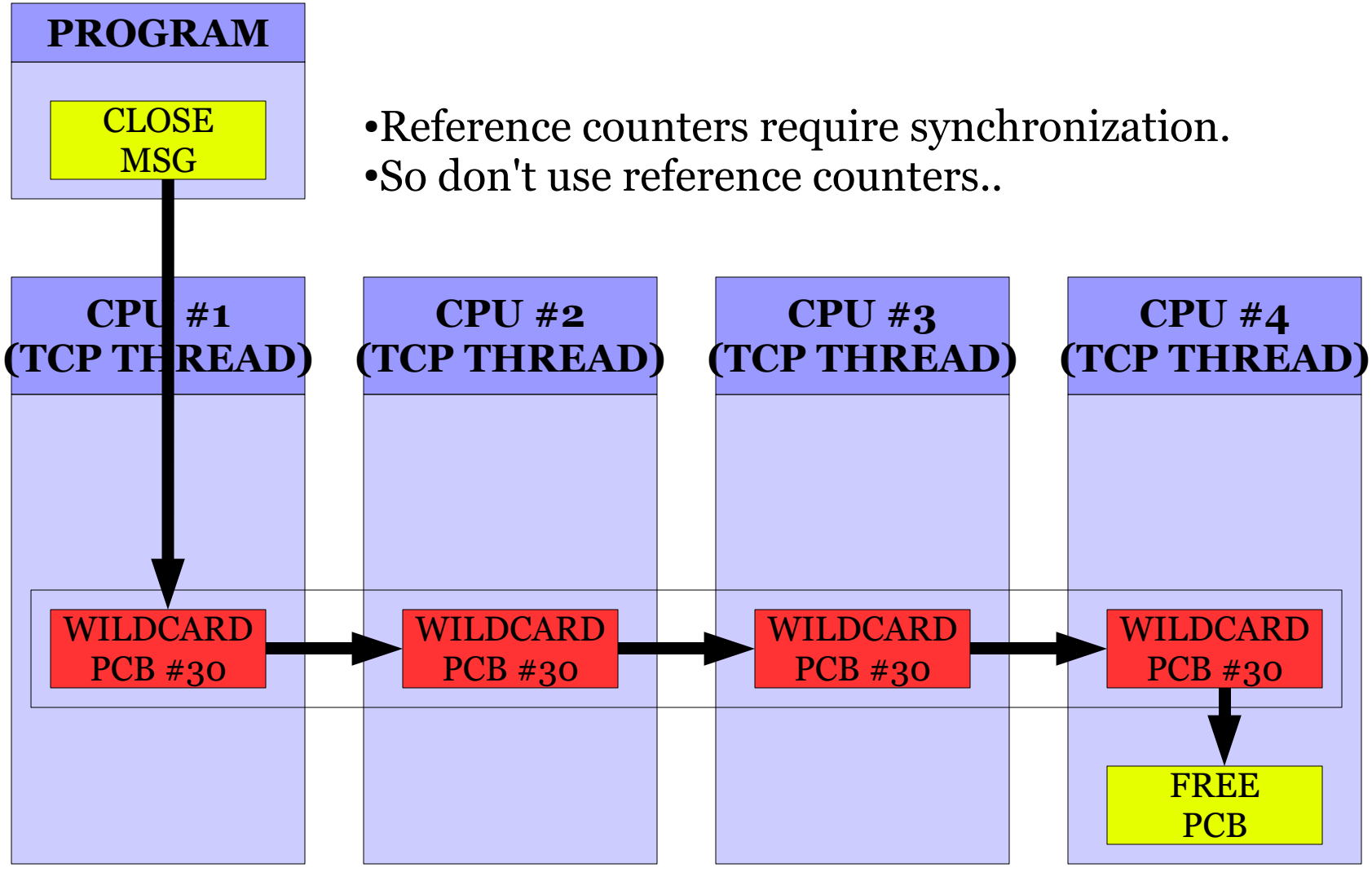

- Mbuf free path expected to cross cpus.
- Cache and return, or cache and reuse?

# Threaded Network Protocol Stacks
# Wildcard PCBs for TCP

- Hash on (srcaddr, srcport, destaddr, destport).
- Replicate wildcard listen sockets.
- Create normal PCB on accept().



# Threaded Network Protocol Stacks
# Wildcard PCBs



# The Namecache

## FREEBSD
optional, disconnected
namespace vnode locked

| **Struct vnode** |
| --- |
| LIST_HEAD(, ncp) v_cache_src;<br>TAILQ_HEAD(, ncp) v_cache_dst;<br>struct vnode *v_dd; |

| **Struct namecache (ncp)** |
| --- |
| LIST_ENTRY(ncp) nc_src;<br>TAILQ_ENTRY(ncp) nc_dst;<br>struct vnode *nc_dvp;<br>struct vnode *nc_vp; |

## DRAGONFLY
mandatory, connected
namespace namecache locked

| **Struct vnode** |
| --- |
| Struct ncp_list v_namecache; |

| **Struct namecache (ncp)** |
| --- |
| TAILQ_ENTRY(ncp) nc_entry;<br>TAILQ_ENTRY(ncp) nc_vnode;<br>struct ncp_list nc_list;<br>struct vnode *nc_vp; |

# The Namecache

- Namespace locked by namecache, not by vnode.
- Negative namecache entries can be created and locked.
- Aliased vnodes do not confuse the kernel.

**VOP_RENAME**

*BSD: VOP_RENAME(dvp, fvp, fcnp, tdvp, tvp, tncp)
DRAGONFLY: VOP_NRENAME(fncp, tncp)

**Struct filedesc**

```
struct vnode *fd_cdir;
struct vnode *fd_rdir;
struct vnode *fd_jdir;
struct namecache *fd_ncdir;
struct namecache *fd_nrdir;
struct namecache *fd_njdir;
```

# The Namecache

- Simplified VFS interface: VOP_NRESOLVE(ncp, cred)
- Simplified kernel interface: nlookup().
- No more namei(), no more lookup()
- No more componentnames, except in compatibility shims.
- Separate NLOOKUPDOTDOT for “..” lookups (ala Linux).
- Topological guarentees for referenced namecache records.
  - Node remains intact in the topology even if file deleted.
  - Traversal to root is guarenteed (getcwd, fstat, and more)
  - Most dvp references replaced by ncp references.
- VFS involvement only for attribute access (for now).

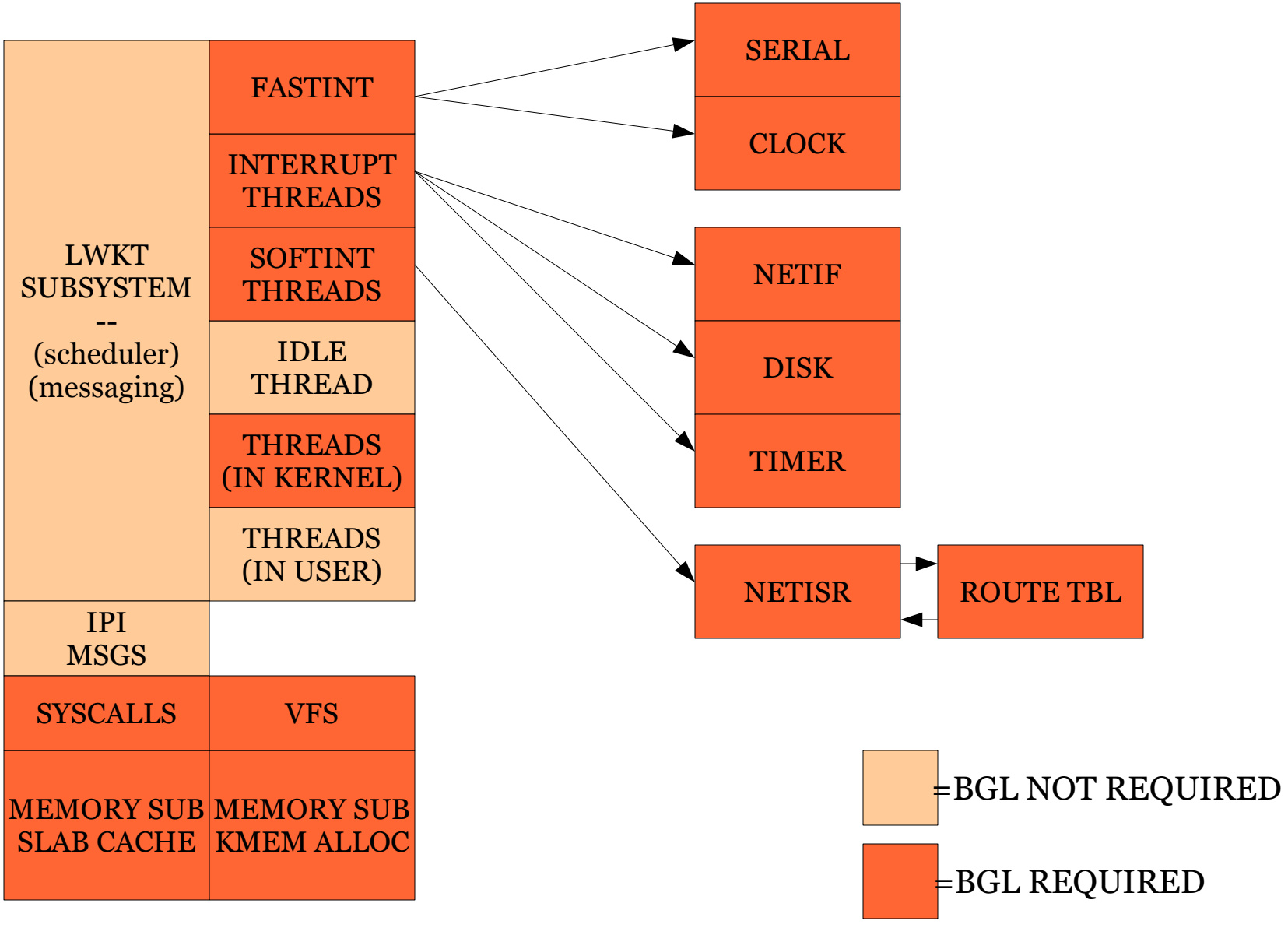
# The Namecache
# Future work

- Intended to be hooked into a cache coherency layer (aliasing, SSI, NFS).
- Direct Attribute Caching.
- Good place for a MAC implementation.
- Clean up VFSs and remove shims supporting obsolete VOP's.

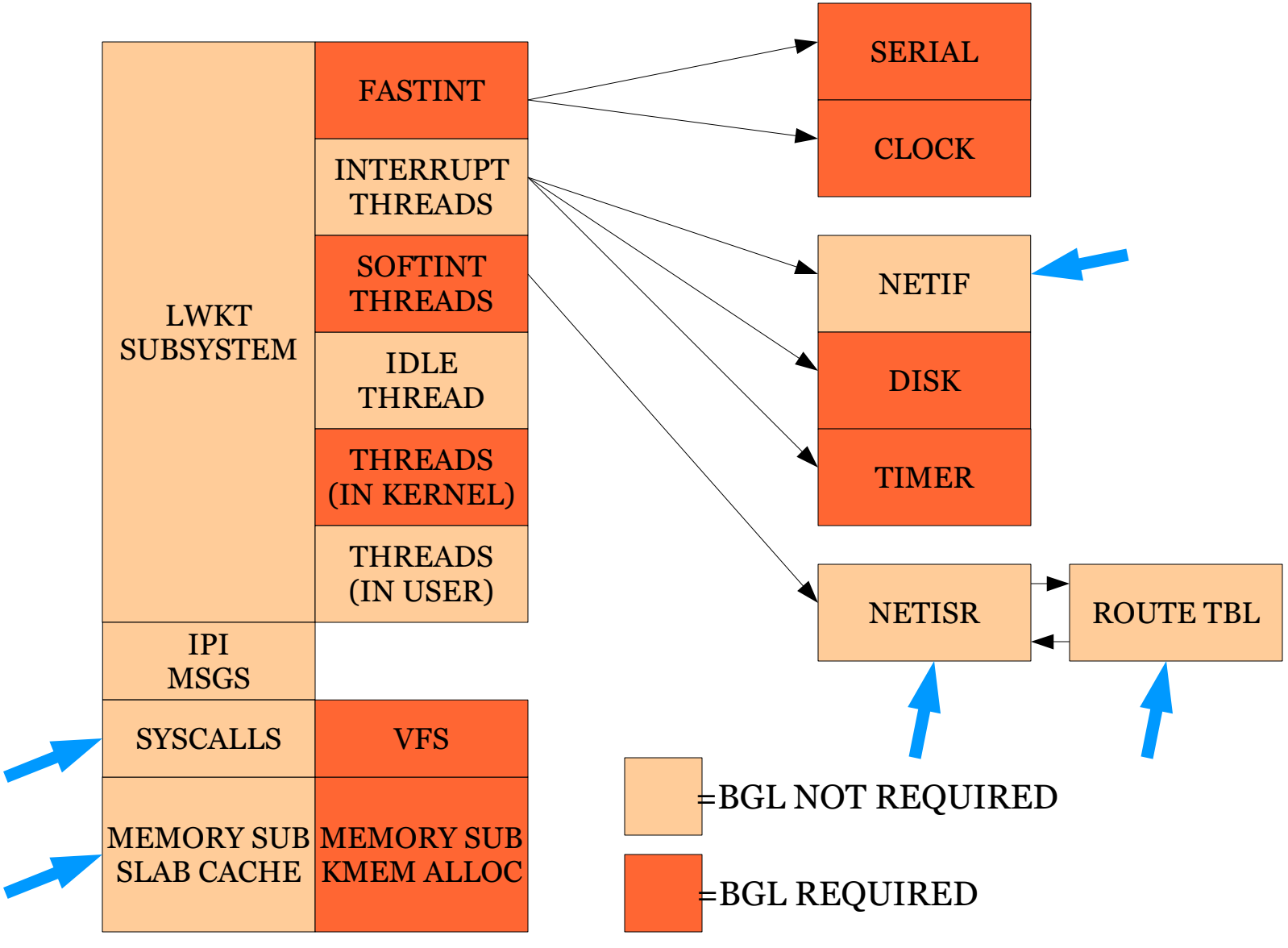
# Big Giant Lock Removal

Matthew Dillon
DragonFly BSD Project
April 2005

# Current BGL Coverage

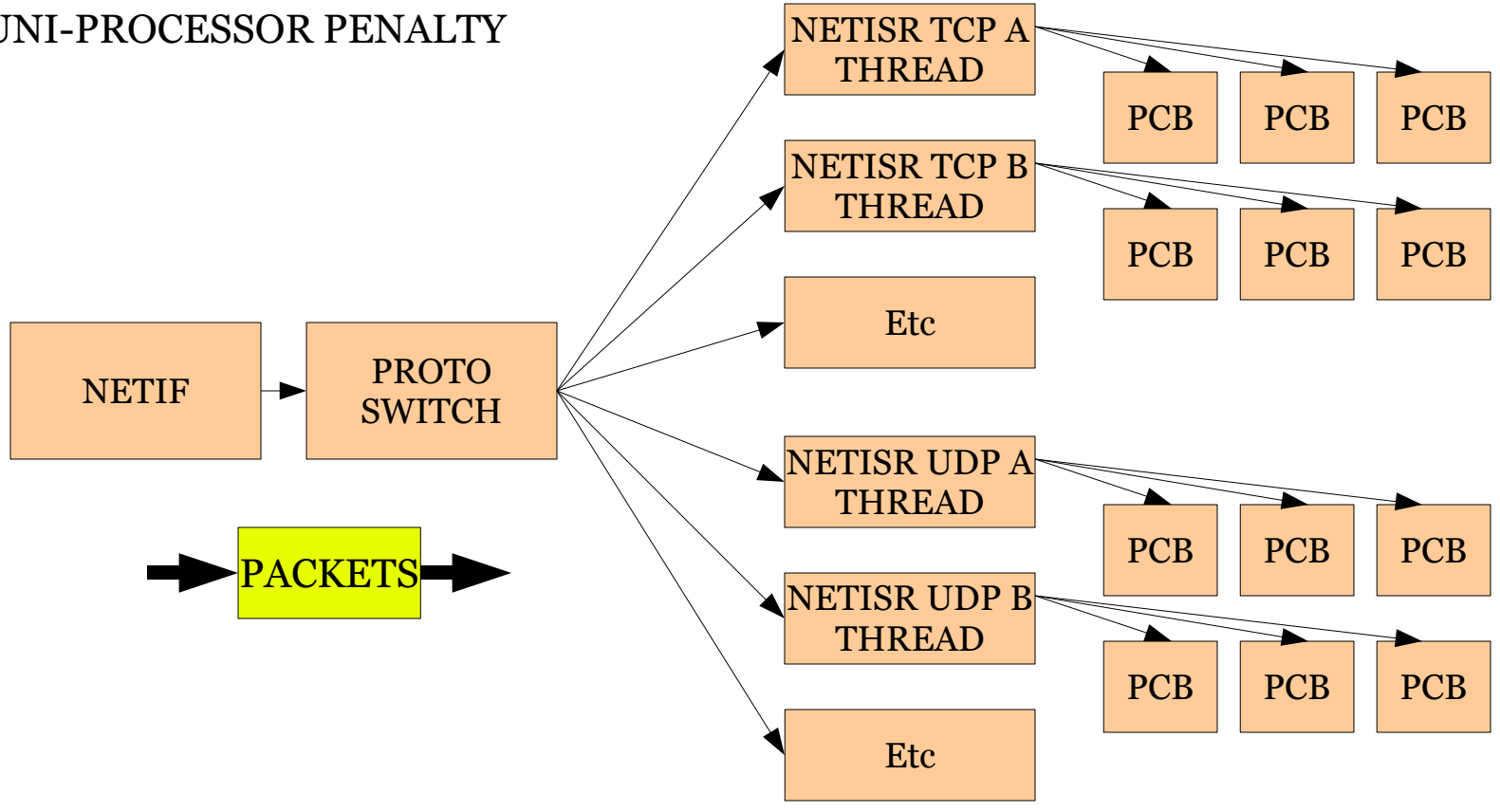


# Next Stage BGL Removal



# BGL Removal – Network Detail

- WORK BEING DONE BY JEFFREY HSU
- UTILIZES LWKT MSGS AND PORTS
- MULTIPLE THREADS PER PROTOCOL
- NO UNI-PROCESSOR PENALTY




=BGL NOT REQUIRED

# The DragonFly BSD Project
# April 2005

Joe Angerson
David Xu
Matthew Dillon
Craig Dooley
Liam J. Foy
Robert Garrett
Jeffrey Hsu
Douwe Kiela
Sascha Wildner
Emiel Kollof
Kip Macy
Andre Nathan
Erik Nygaard
Max Okumoto
Hiten Pandya
Chris Pressey
David Rhodus
Galen Sampson
YONETANI Tomokazu
Hiroki Sato
Simon Schubert
Joerg Sonnenberger
Justin Sherrill
Scott Ullrich
Jeroen Ruigrok van der Werven
Todd Willey

and Fred -->


WWW.DRAGONFLYBSD.ORG