Network working group

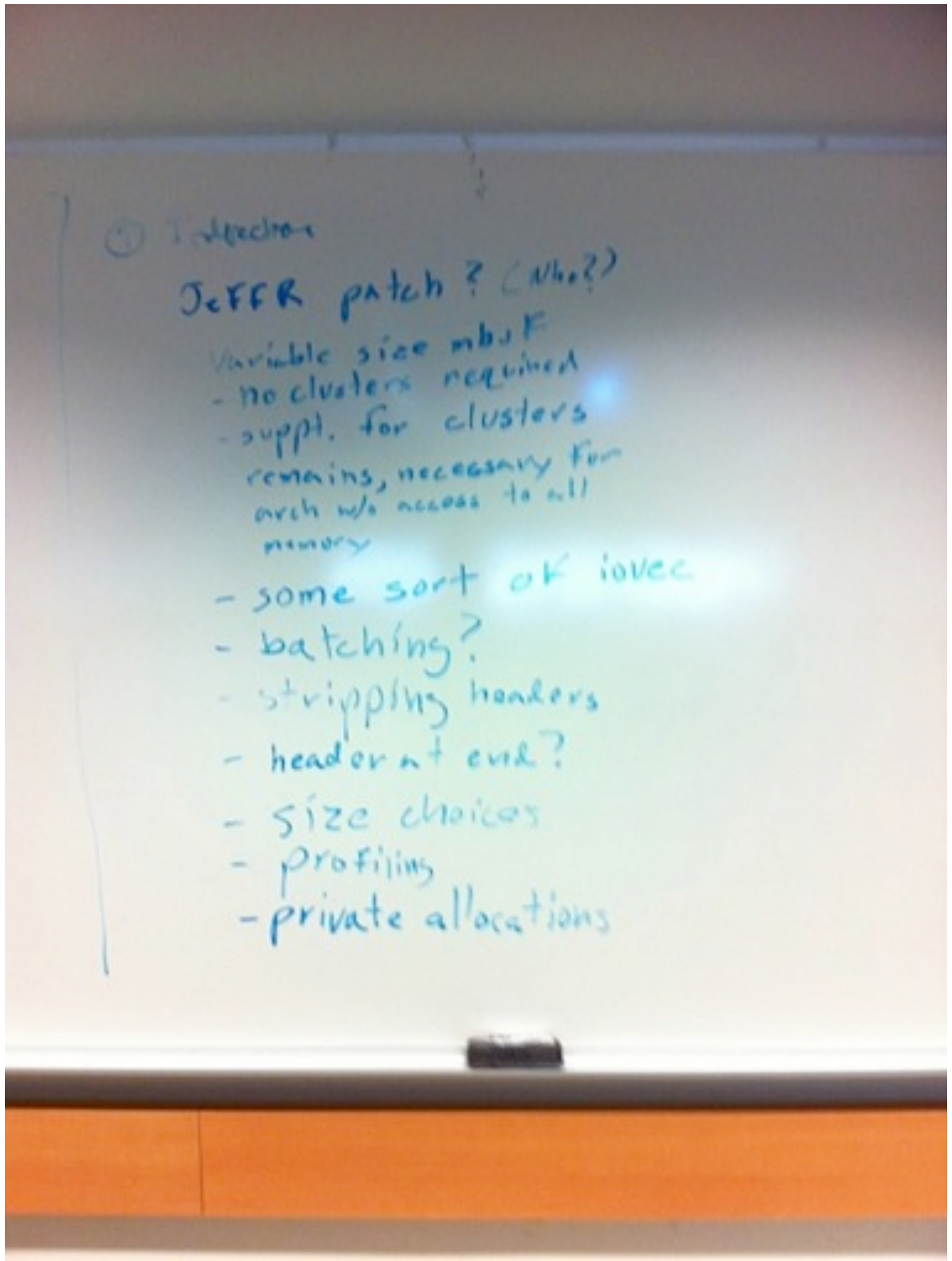
Robert N. M. Watson
University of Cambridge (etc)

Agenda

- struct ifnet
  - L2/L3 redesign -- decompose
  - Indirection reduction
  - Multiqueue visible at ifnet
  - De-duplicate 10gbps driver infrastructure
- struct mbuf
  - Variable-size mbufs
  - Meta-data facility
  - Indirection reduction
- And lots of other topics we didn’t have time for

Variable-size mbufs

- Jeff Roberson prototype two years ago
- Allow mbuf size to vary to avoid unnecessary cluster indirection
    - Modern CPU designs dislike pointers
    - Retain external storage for sendfile(), etc
- Concerns
    - How to handle “private allocators”, NUMA, etc.
    - Indirection shift for batched packets
    - Profiling required -- especially, packet size distribution
- Consensus that this is a good idea



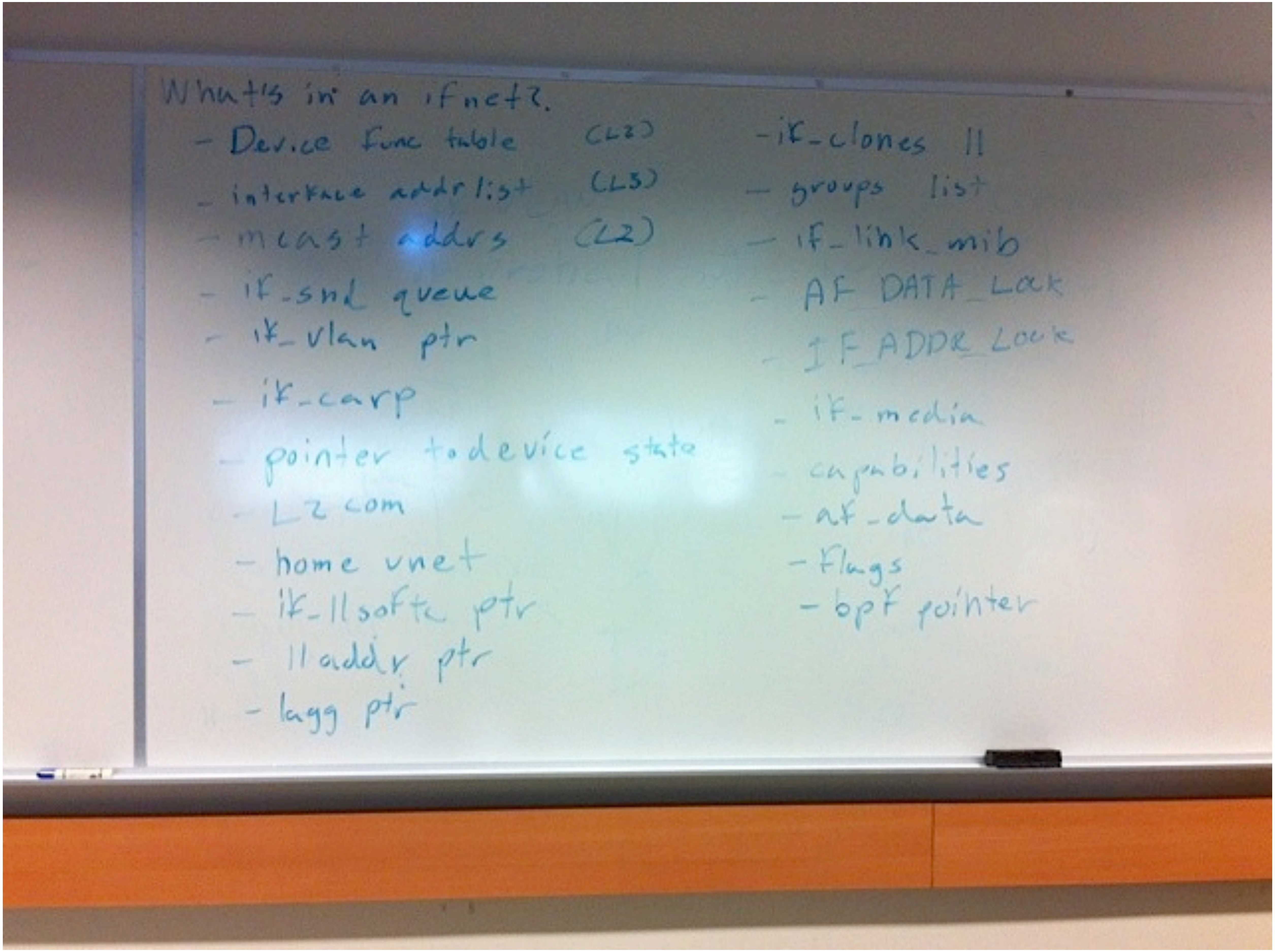
① Indirection

JeFFR patch? (Who?)

Variable size mbuF
- no clusters required
- suppt. for clusters remains, necessary for arch w/o access to all memory
- some sort of iovec
- batching?
- stripping headers
- header at end?
- size choices
- profiling
- private allocations

L2/L3 redesign

- Clarify design, reflect changing reality
  - E.g., 802.1 VLAN vs ethernet confusion
- Decompose ifnet by layer
  - Logical vs. physical vs. protocol attachment
- Scalability requirements: 10,000,000 “subscribers”
- Juniper design and Adara interests both relevant

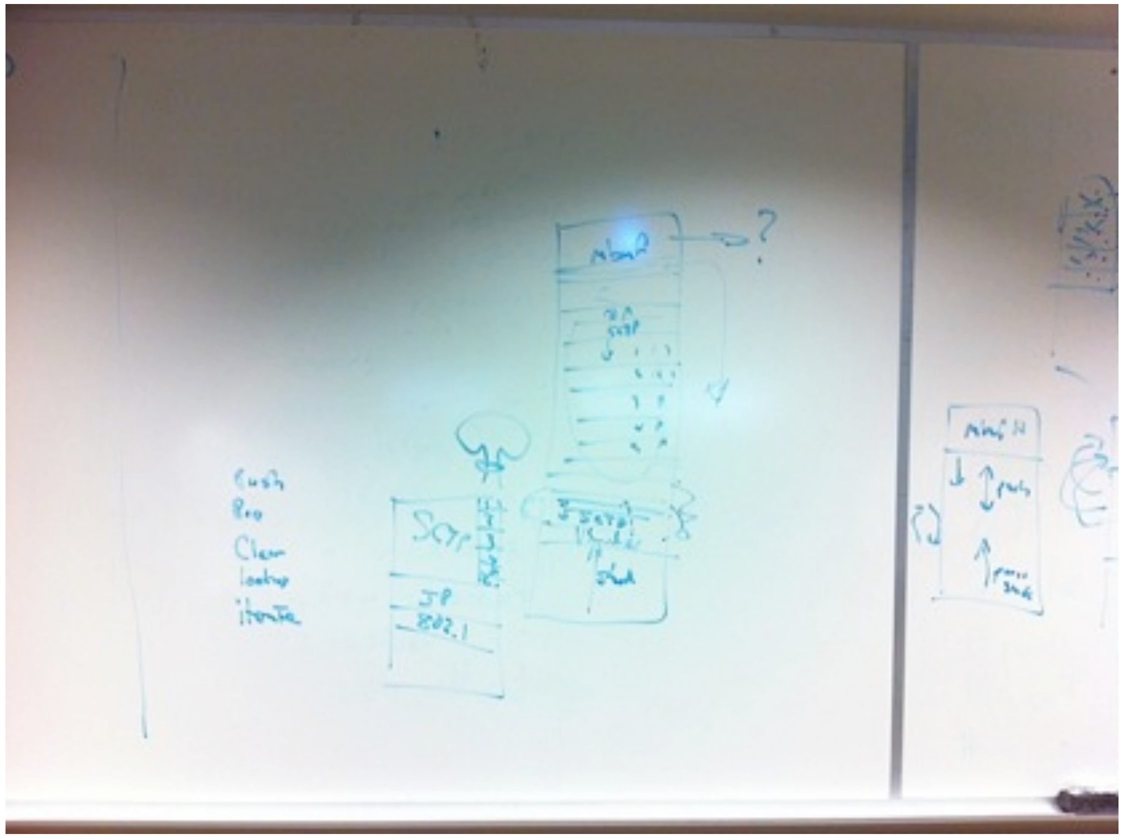


What's in an ifnet?.

- Device func table (L2)
- interface addr list (L3)
- mcast addrs (L2)
- if_snd queue
- if_vlan ptr
- if_carp
- pointer todevice state
- L2 com
- home vnet
- if_llsoftc ptr
- lladdr ptr
- lagg ptr
- if_clones ll
- groups list
- if_link_mib
- AF_DATA_LOCK
- IF_ADDR_LOCK
- if_media
- capabilities
- af_data
- flags
- bpf pointer

mbuf meta-data

- Two current models
  - Fixed mbuf headers: very fast, constrained
  - m_tags: quite slow, very constrained
- Growing consumer set
- Different types of meta-data
- Tension between parsing/ rewrite and decapsulation
  - Middle nodes vs. edge nodes
- Hand wave at flexible embedded tags and stacks

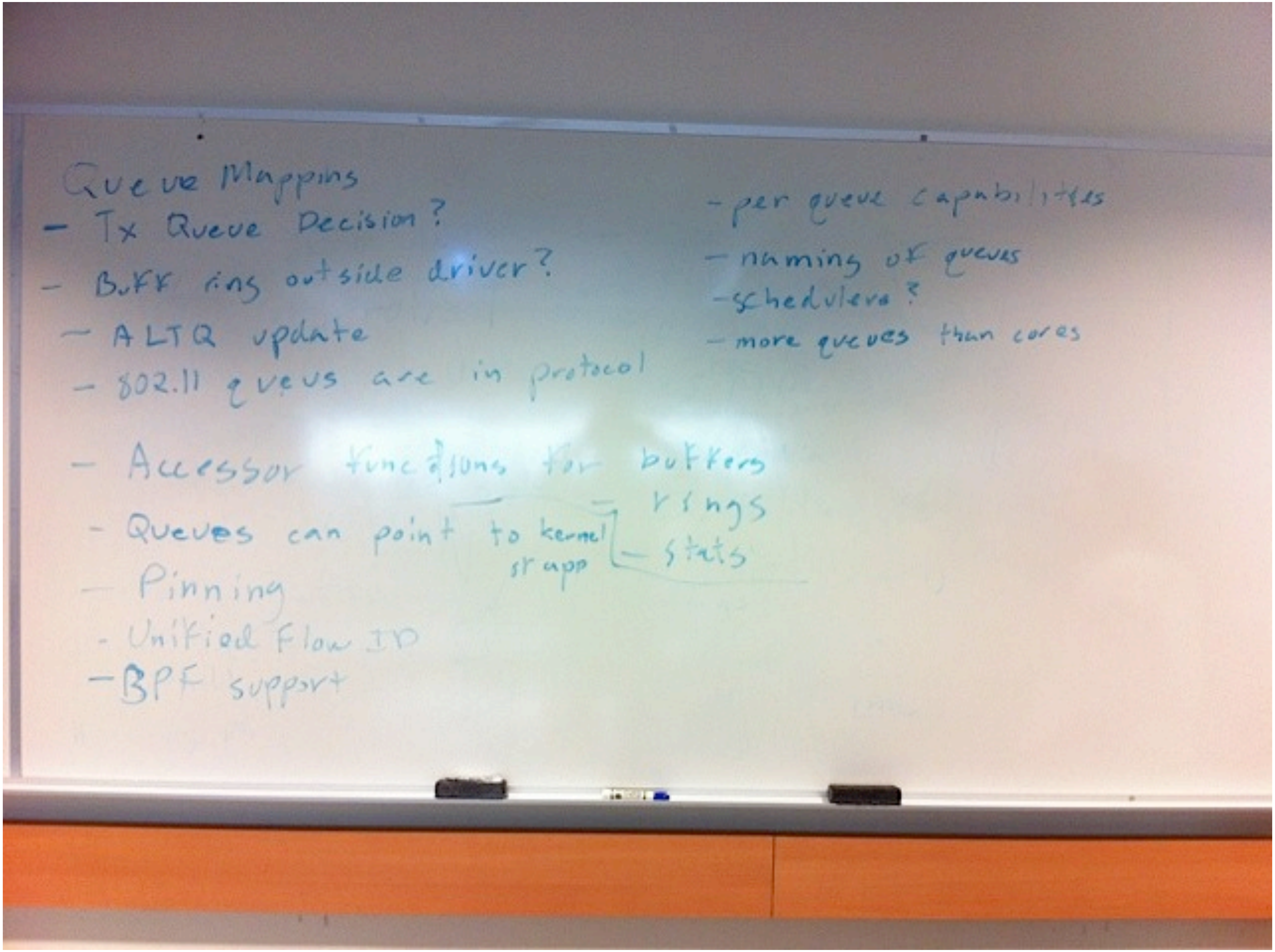


mbuf meta-data

- Two current models
  - Fixed mbuf headers: very fast, constrained
  - m_tags: quite slow, very constrained
- Growing meta-data consumer set
- Different types of meta-data
- Tension between parsing/rewrite and decapsulation
  - Middle nodes vs. edge nodes
- Hand wave at flexible embedded tags and stacks

Multiqueue

- Device drivers internalise queue logic
- Want to pull up (down) stack
  - Management, statistics, BPF
  - Allow stack integration, especially on transmit
  - Scheduler integration
- Concerns
  - But who does “queue” belong to? 802.11 vs. 802.1 vs … Cores vs. threads vs …
  - Interactions with L2/L4 rewrite



Queue Mappins

- Tx Queue Decision?
- Buff ring outside driver?
- ALTQ update
- 802.11 queus are in protocol
- Accessor functions for buffers
  - rings
  - stats
- Queues can point to kernel or app
- Pinning
- Unified Flow ID
- BPF support

- per queue capabilities
- naming of queues
- schedulers?
- more queues than cores

Conclusion

- Consensus to move forward on projects
  - L2/L3 rewrite / ifnet reconstitution
  - Multiqueue exposure to stack
  - Variable-size mbufs
- Consensus on common interest
  - Meta-data facility