

# The x-Kernel: An Architecture for Implementing Network Protocols

Norman Hutchinson and Larry  
Peterson, UA

# XKernel - objective

- Other kernels
  - protocols separate processes
  - or modules, without clean/uniform interfaces
  - a lot of typical protocol operations re-invented in each protocol – separate buffer management, identifier management...
  - top level abstraction, i.e., socket – uniform, but doesn't easily translate into lower level protocols
    - e.g., fine if on top of TCP, not so if on top of Ethernet directly...
  - Finally, can introduce a new optimized protocol in kernel, or build kernels specialized for some type of communication (e.g., RPC), but no flexibility to generalize to other protocols
- X-Kernel:
  - specifies interfaces b/w protocol layers, and b/w protocols and upper level software
  - process per message – avoids copy, context switches, promotes concurrency, flexibility for different comm. requirements
  - supports key mechanisms needed for building protocols
  - doesn't sacrifice efficiency over custom optimized protocols
  - first arch for protocol composition

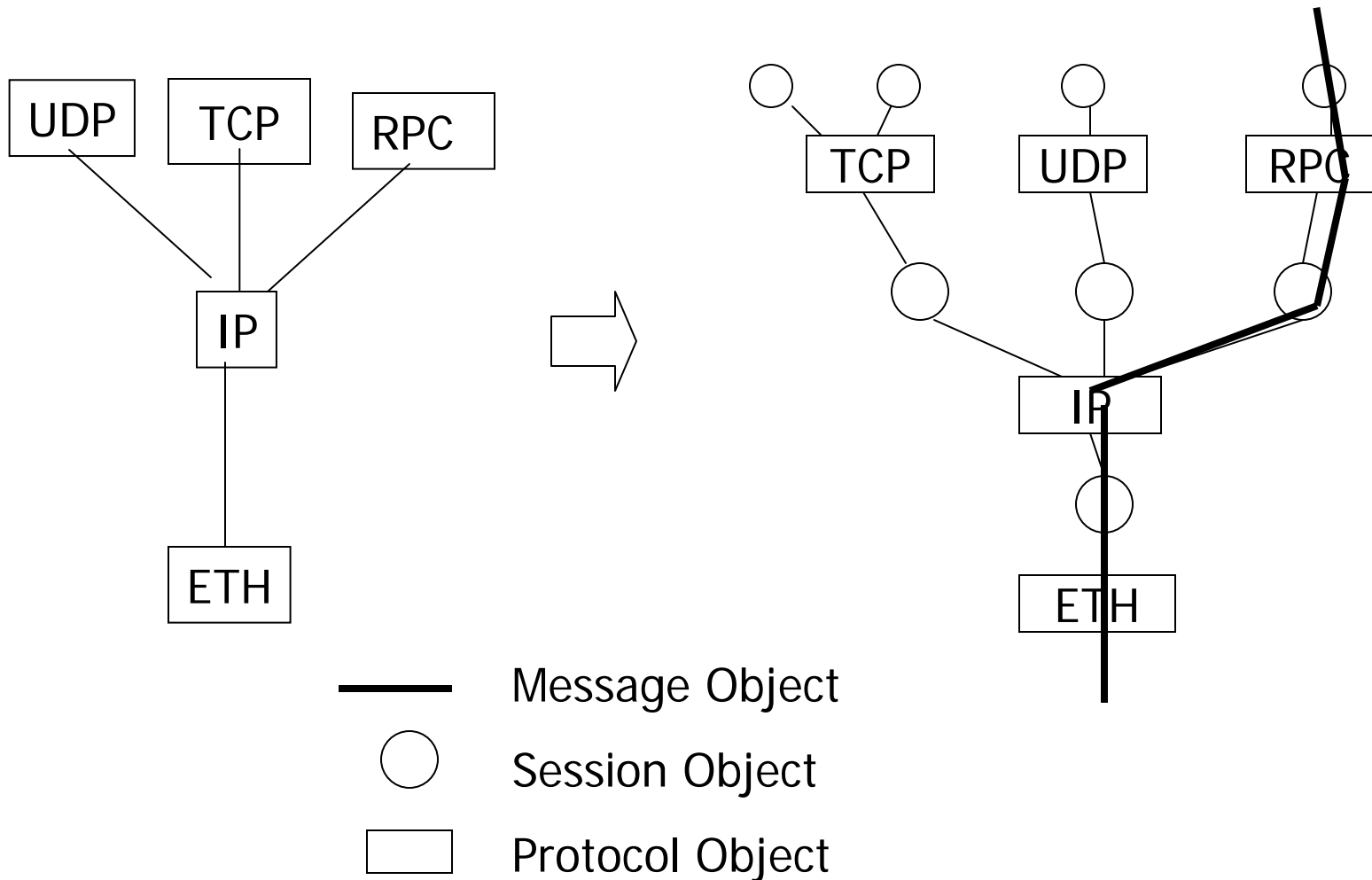
# Architecture

- Multithread kernel; trap for user-kernel, upcall for kernel-user
- Ops for buffer management (tail, header, split, merge, multiple refs...), map manager (for IDs), event mgr (for timers)
- OO approach, in C, obj have access to other stacks

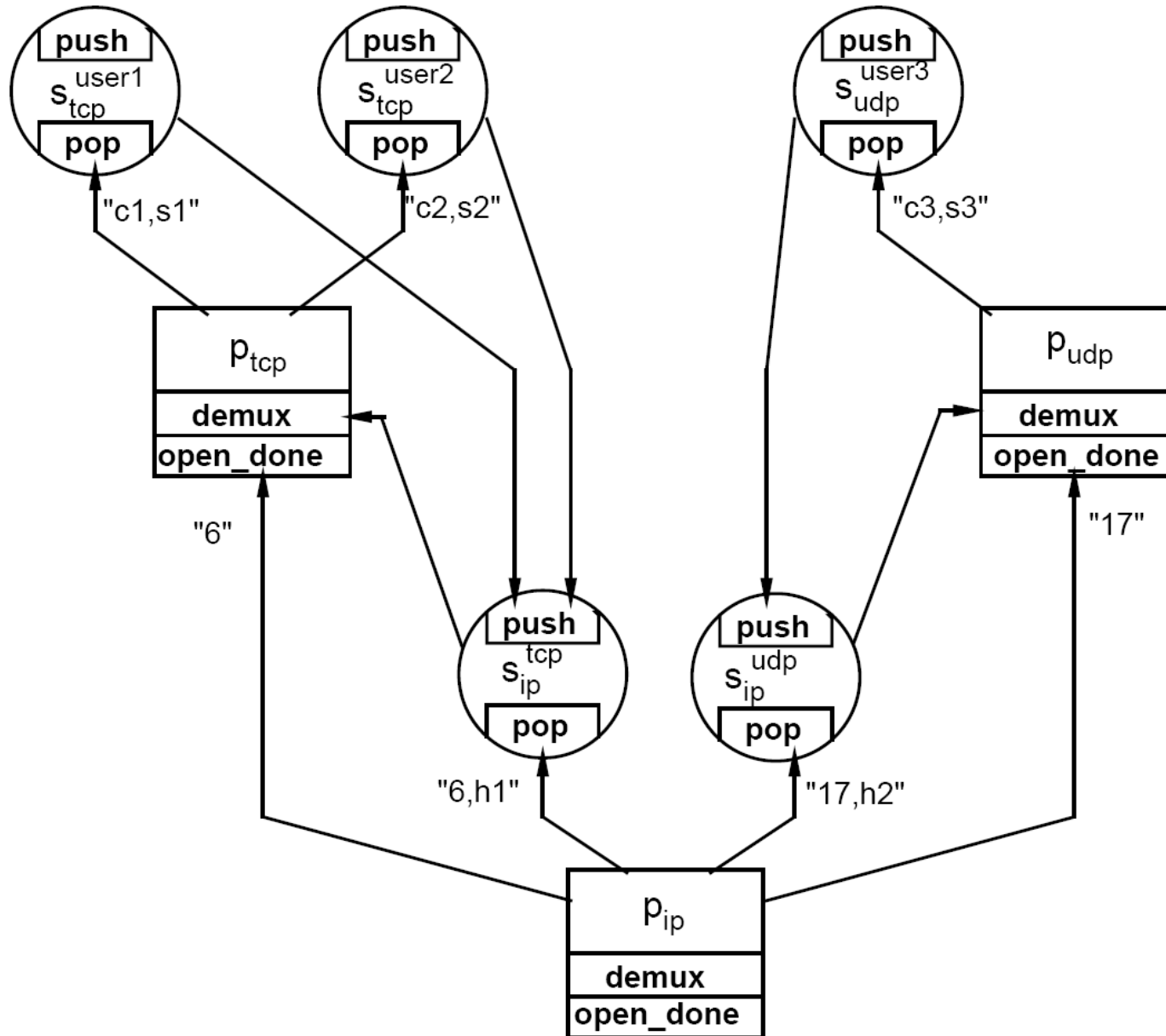
## Abstractions:

- Protocols – passive; ops: open, open\_enable, open\_done & demux
- Sessions – passive; push & pop
- Messages – active; a kernel thread is dispatch it while visiting protocol and session obj.; no ctx switches unless msg must wait in a session object
- User – protocol\_create

# X-kernel Configuration



# Example



# Performance data

Component	Percentage
Buffer Manager	21.8
Id Manager	1.8
Ethernet	43.7
IP	9.8
UDP	2.8
Interface Overhead	5.3
Boundary Crossing	5.9
Process Management	8.6
Other	0.3

Protocol Stack	<i>x</i> -Kernel (msec)	Unix (msec)
ETH	1.68	4.87
IP-ETH	1.89	5.62
UDP-IP-ETH	2.00	5.36
TCP-IP-ETH	3.30	6.10
RPC-UDP-IP-ETH	4.00	9.20

Table III: User-to-User Latency

# User-to-user bw

Protocol	<i>x</i> -Kernel (k-bytes/sec)	Unix (k-bytes/sec)
UDP	528	391
TCP	417	303

# Supporting routines

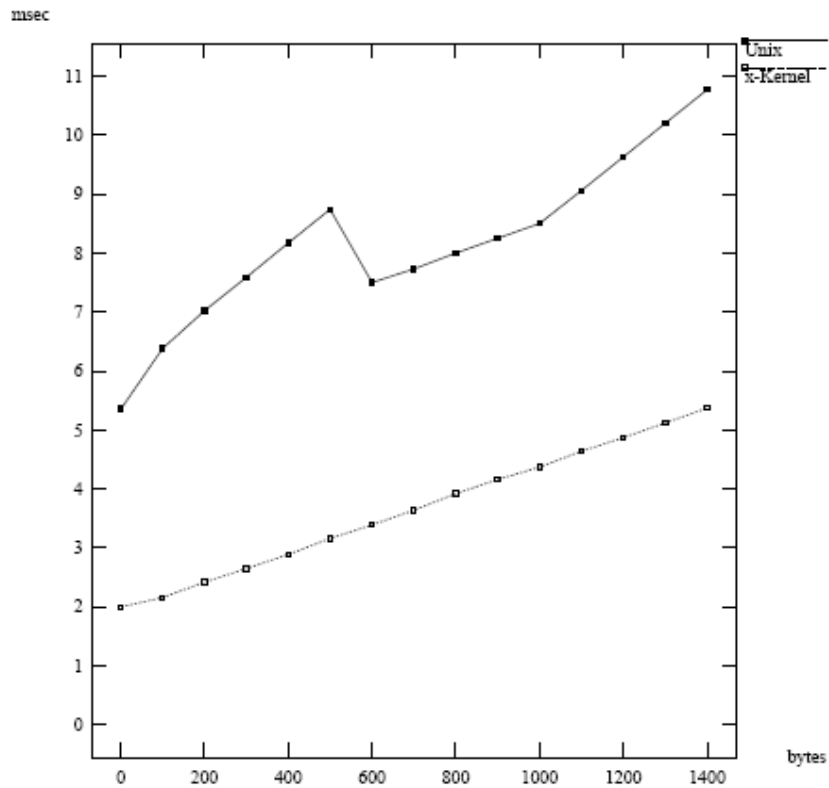


Figure 5: Message Manager

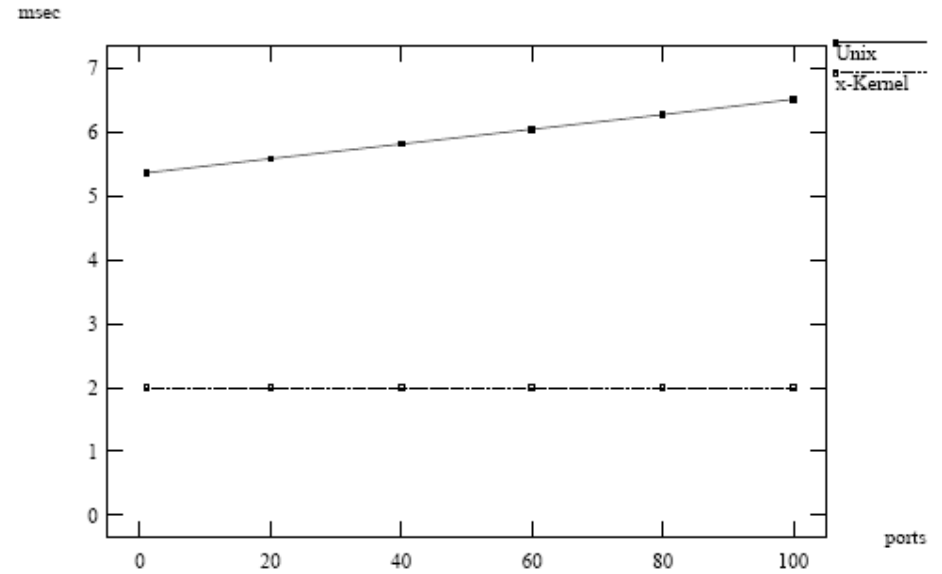


Figure 6: Map Manager