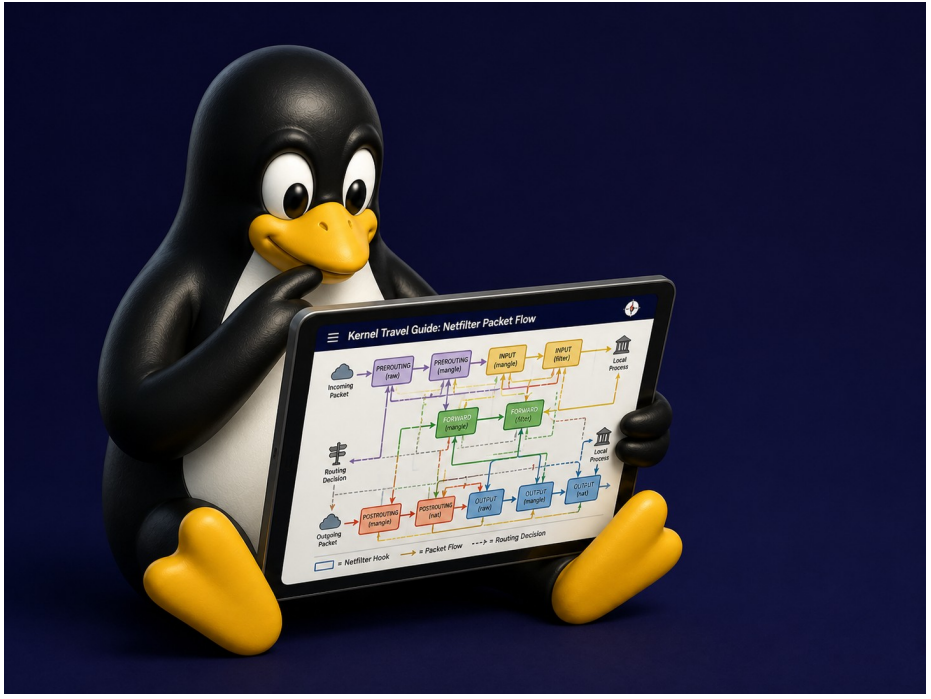
DPDK SUMMIT STOCKHOLM
12 MAY 2026

Finding the Best Path to the Kernel

Stephen Hemminger
<stephen@networkplumber.org>

Roadmap

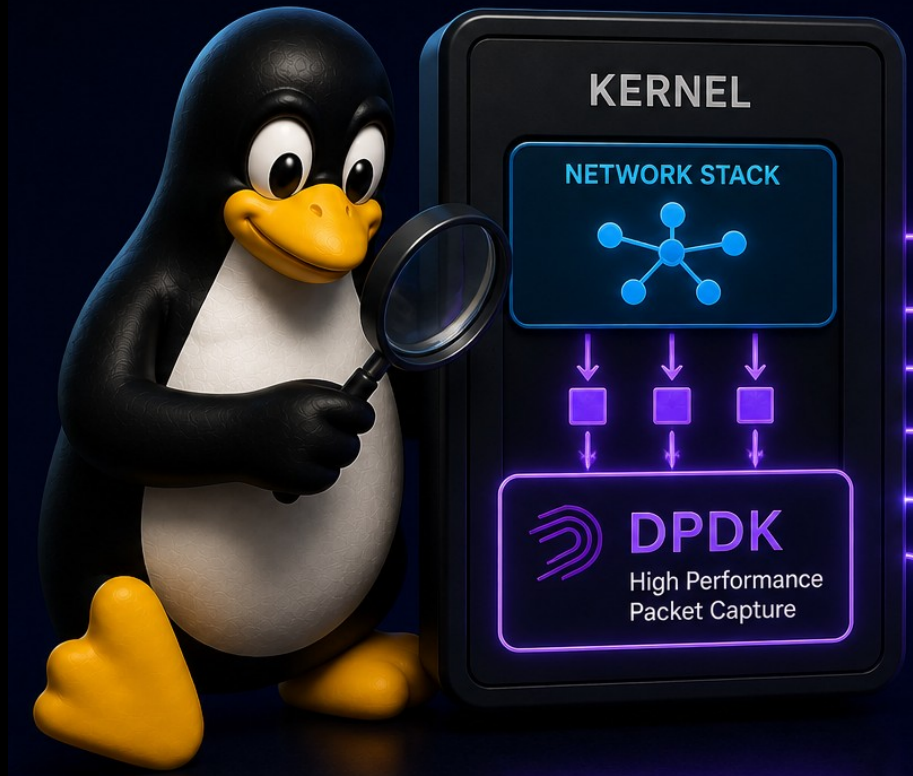
- Why so many flavors?
  - How do they differ
- What is the best one?
  - Construct a mini-testbed
- Future ideas



What? – DPDK Tools

- Decision Fatigue
- Choice matters
- Confusing

Redirection Use Cases



Selective Acceleration
Accelerate critical traffic while the rest stays in the kernel

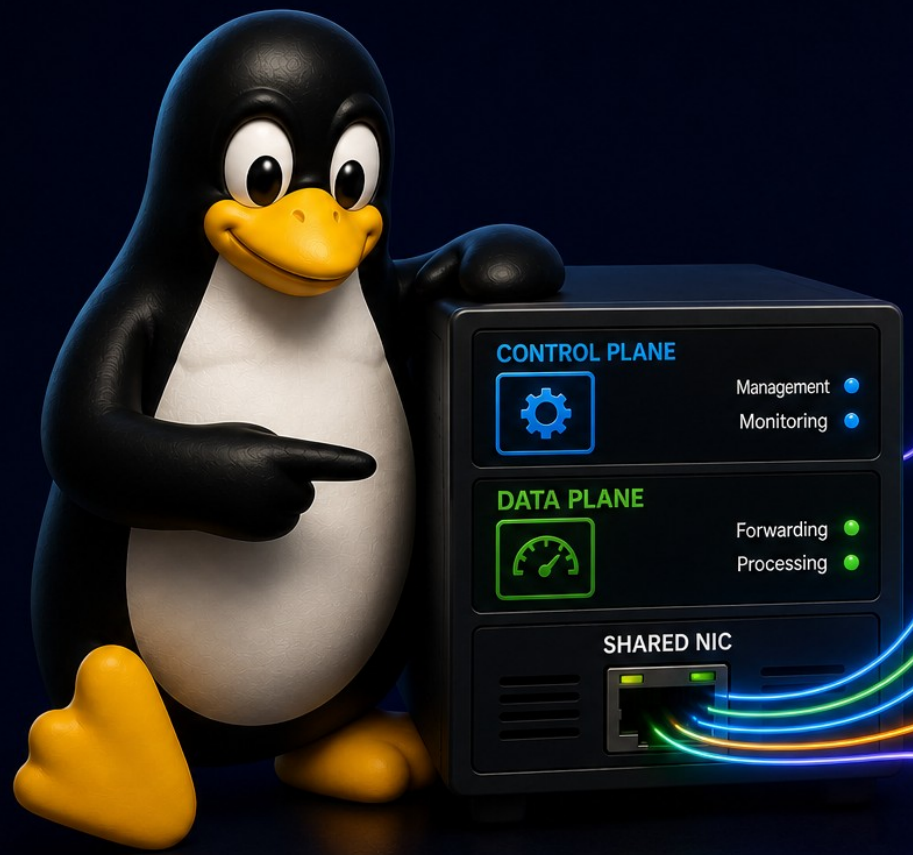
Cloud & Containers
Efficient packet capture in virtualized and container environments

Traffic Inspection
Capture packets for IDS, monitoring, and analytics

Development & Testing
User-space packet access for faster iteration and experimentation

Unsupported Hardware
Enable high-performance packet capture on NICs without kernel support

Injection Use Cases



Shared NIC
Efficient use of a single network interface

Control and Dataplane on Same Device
Simplified architecture, lower cost

Network Middlebox (Appliance)
Firewall, NAT, Load Balancer, Proxy, etc.

Virtual Private Networks
IPsec, WireGuard, OpenVPN

Intrusion Detection
Inspect and analyze network traffic

Remote Access
Secure administrative and user access

Two Models

Redirect (Bifurcation)

- Kernel controls NIC
- DPDK takes packets

Examples:

- AF_PACKET
- AF_XDP

Injection

- DPDK owns NIC
- DPDK gives packets

Examples:

- TAP
- Virtio User

Key differences

- Access to kernel
  - System calls read/write to send/receive
  - Shared memory ring buffer
- Data copies and Ownership
- Polling vs Wake-up
- Userspace vs Kernel threads

	Inject (kernel receives packets)			Redirect (socket sees packets)	
Aspect	TAP	rTAP	virtio-user	AF_PACKET	AF_XDP
Syscall API	readv / writev	io_uring	shared memory	TPACKET_V2 mmap	UMEM mmap
Ring buffer	none	io_uring + fd	virtio ring	fixed-slot	UMEM rings
Checksum offload	SW	HW	HW	no	no
TSO	no	yes	yes	no	no
Timestamps	none	none	DPDK-internal	software	xsk metadata
Kernel thread	softirq	softirq	vhost kthread	softirq	softirq + NAPI
Wakeup	syscall	io_uring SQE	eventfd kick	poll / syscall	need_wakeup

Faster TAP: rTAP experiment

- TAP device limited
  - Emulation of checksum and TSO
  - Per packet read/write
- RTAP experiment
  - New Linux Async API – io_uring
  - Uses kernel for checksum offload

Is it faster? Let's see

Benchmarking Kernel Injection

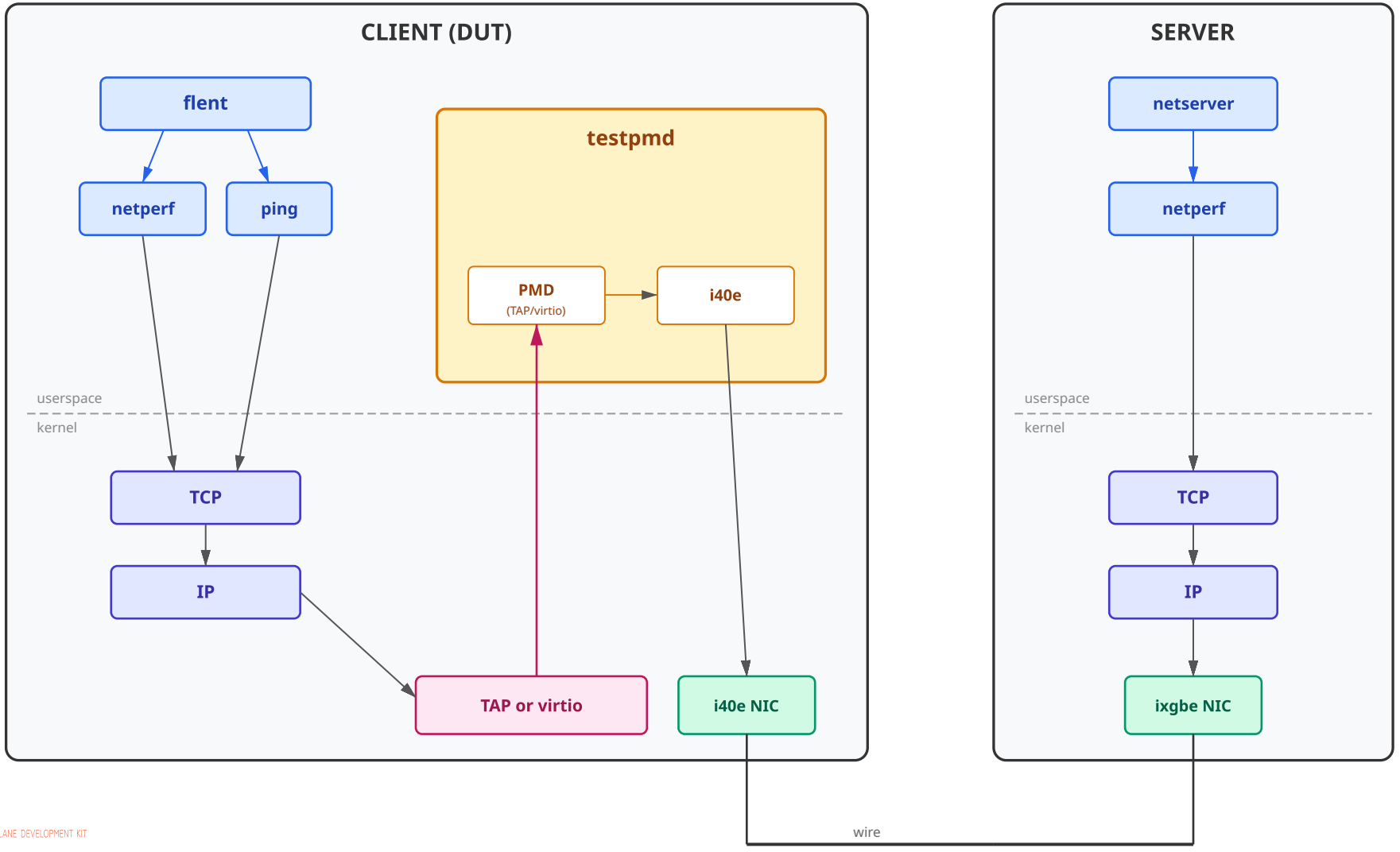
Hardware

- Server
  - Arm Cortex-A720 12 core
  - Debian old-stable (bookworm)
  - 10G ixgbe
- Client DUT
  - Intel i9-13900H 14 core
  - Debian stable (trixie)
  - DPDK 26.03
  - 10G i40e

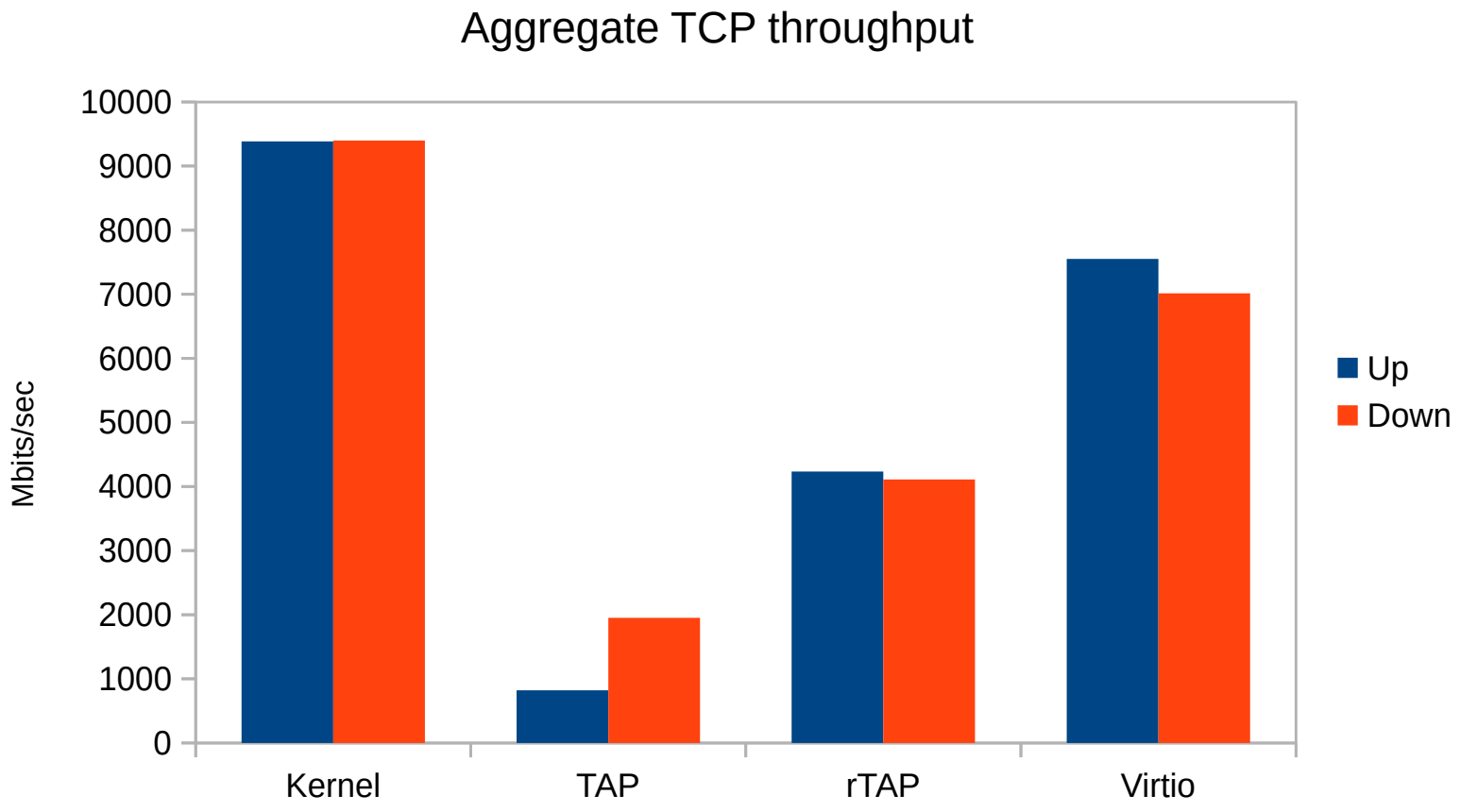
Software

- Flexible Network Tester (flent)
- Realtime Response Under Load (rrul)

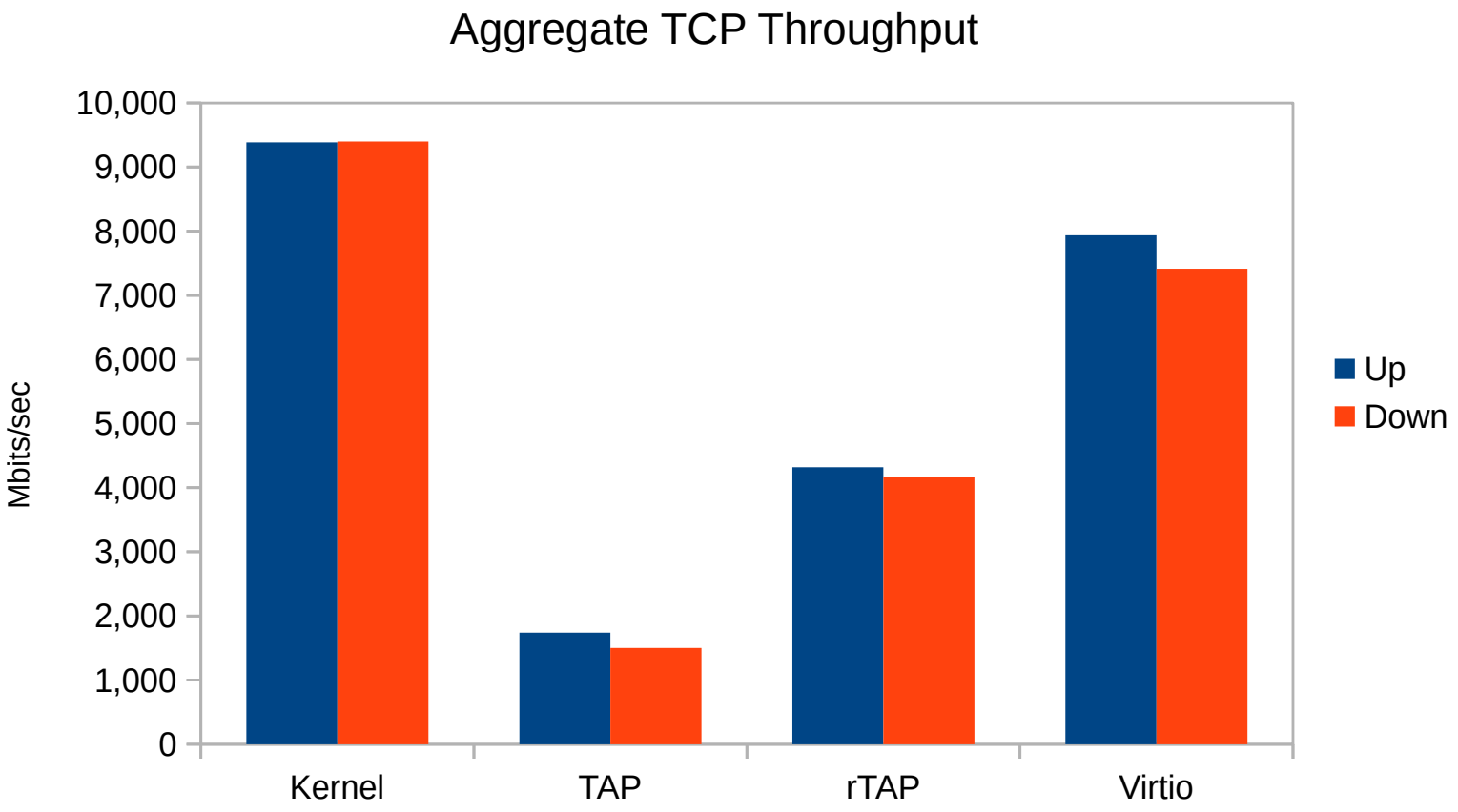
Realistic (non-trivial) but not full blown iMix or WebBench


CLIENT (DUT)
flent
netperf
ping
testpmd
PMD
(TAP/virtio)
i40e
userspace
kernel
TCP
IP
TAP or virtio
i40e NIC
SERVER
netserver
netperf
userspace
kernel
TCP
IP
ixgbe NIC
wire

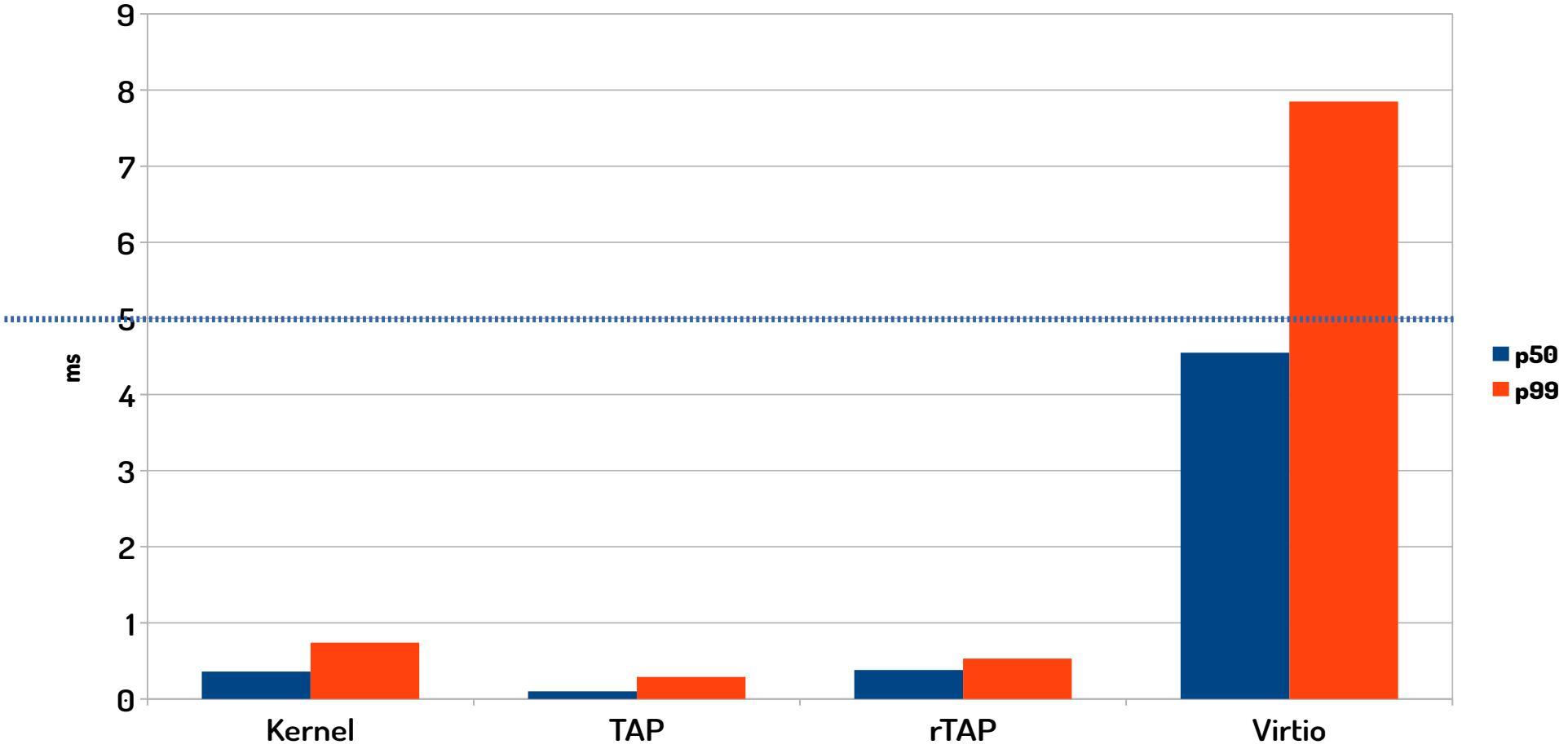
Benchmark - no offloads



Benchmark - checksum offload



Latency under load

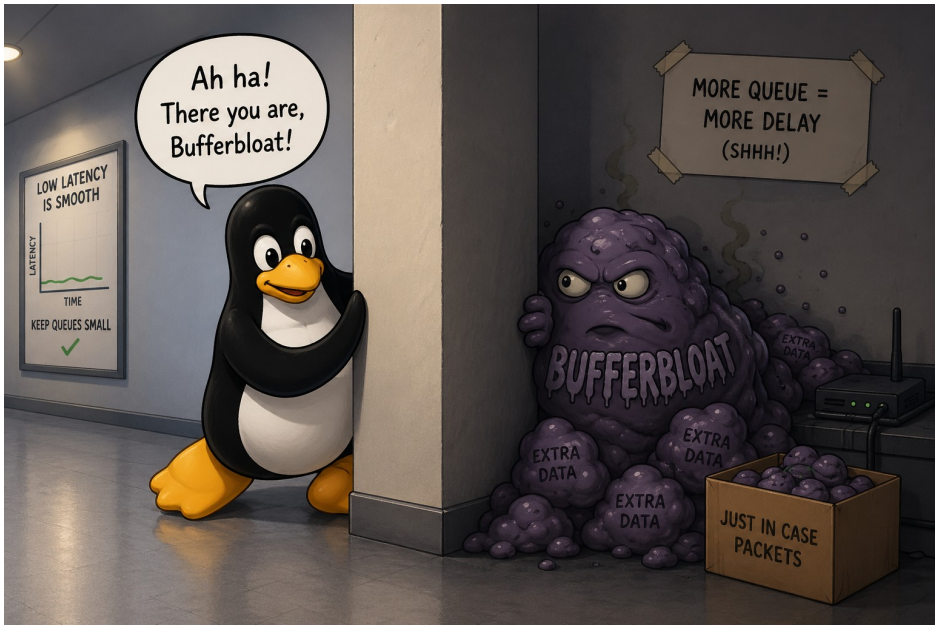


Problems

- Testpmd
  - Manual configuration
  - Does not adapt to offloads
  - Using checksum requires data hit
  - Perf shows test-pmd is the hotspot
- TCP Segmentation Offload
  - Hardware limitation
  - Driver bugs
  - Queuing Delays

1024 TxD * 64kB = 64 MB = 53ms at 10G!

What is lurking here?



LPC 2014 talk

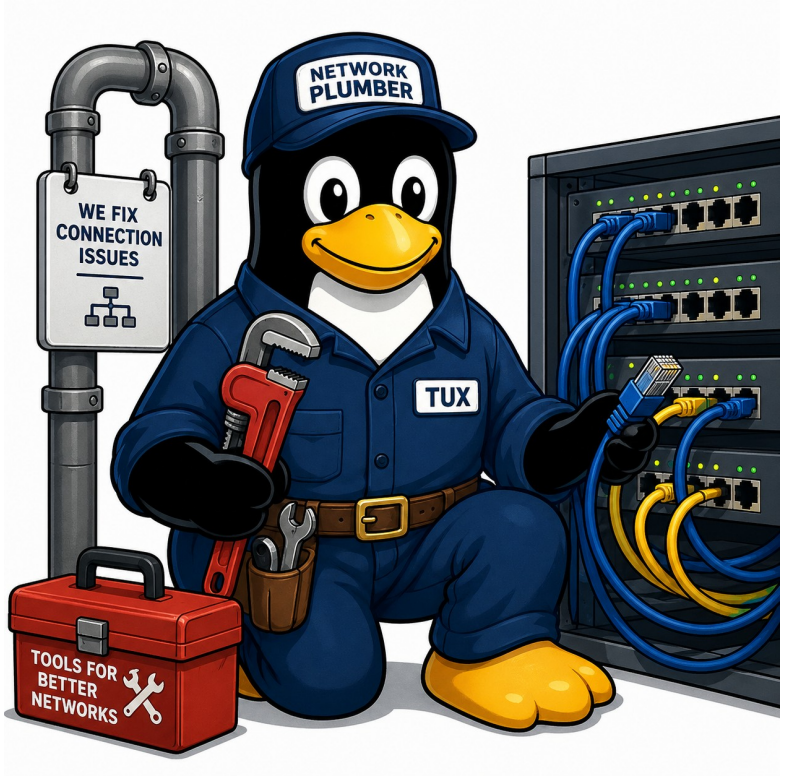
Missing PMD's on non Linux

- FreeBSD
  - TAP – diverged from Linux
  - /dev/bpf – is equivalent of AF_PACKET
- Windows
  - TAP - 3rd party driver from OpenVPN
  - Npcap – driver used by libpcap

Future Example - l2fwd-kernel

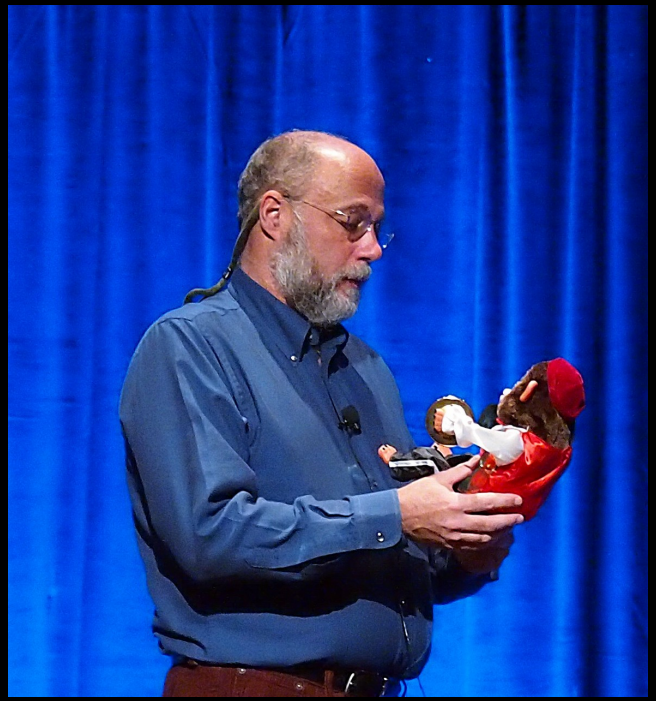
- Adapts to hardware
  - No nerd knobs; just works
- Tracks kernel control path (netlink)
  - Link up/down; mtu; ethtool etc
- Fix Bufferbloat
  - BQL? Codel? Cake?

Thank you



Finding the Best Path to the Kernel

Dedication



Dave Täht 1965 – 2025

lwn.net/Articles/1016109