THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

WAND
University of Waikato
Network Research Group

openLI

Packet Capture for LI

OpenLI Training: Chapter Three

Shane Alcock
University of Waikato
New Zealand
shane.alcock@waikato.ac.nz

Packet Capture

- Core activity of your LI system
  - Customer communications traverse network as packets
  - “Capture” the packets for the intercept targets
  - Encapsulate, then mediate

pcap

- All operators should be familiar with pcap and `tcpdump`
  - Simple capture format
  - Available everywhere
  - Read from live interfaces or saved files on disk (pcaps)
  - Write to a file on disk

pcap

- We already know `tcpdump` isn’t suitable for LI
    - The main concern is the output format

- libpcap
    - Underlying library that implements pcap capture
    - `tcpdump` is built on top of libpcap
    - Other libpcap tools exist
        - `tcptrace, tcpslice, snort`
    - Why not write a libpcap tool for interception?

The Key Issue: Performance

- An LI system must not drop packets
    - Packet capture is an obvious potential culprit
        - Small buffers
        - Low priority compared to other tasks

- LI system design must optimise for best capture performance
    - The goal of this lesson!

Things That Affect Performance

- Incoming packet rate
    - Each individual captured packet requires processing effort
        - Even ignoring packets takes some effort
    - Pre-capture filtering can make a huge difference
    - Configure devices to only mirror relevant traffic to capture

Things That Affect Performance

- Packets per second is much more important than Gbps
  - 10Gbps @ 1500 byte packets == ~820 Kpps
  - 10Gbps @ 64 byte packets = ~14,881 Kpps
  - **18x** the workload between best and worst case

Things That Affect Performance

- Ability to process in parallel
  - Spread processing load across multiple CPU cores
  - Requires support from the capture method

Things That Affect Performance

- Kernel networking stack is designed for general use
  - Interrupts and system calls introduce overheads

Things That Affect Performance

- Bypass the kernel to achieve more speed
  - Specialised APIs to pass packets directly into userspace
  - Only applications that support the API can be run

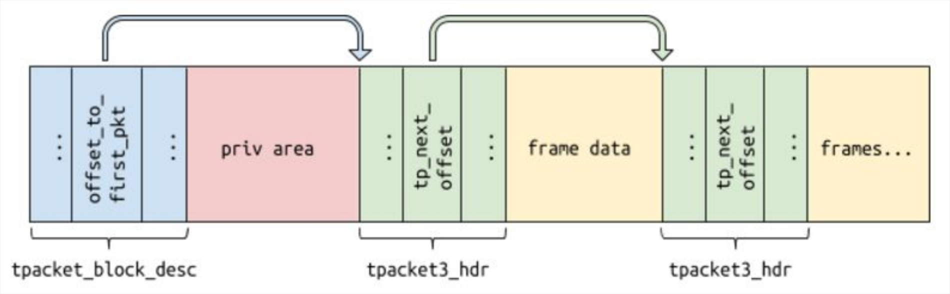
pcap Revisited

- Not ideal for high performance capture
  - User space library, no kernel bypass
  - Not well-suited to streaming packets in parallel
  - Strong likelihood of capture loss at high packet rates

- Fine for testing and troubleshooting
  - Not for production LI systems on modern networks

AF_PACKET sockets

- Linux native capture method
  - Uses a ring buffer to store observed packets in physical memory
  - Buffer is mapped into virtual memory for user access
  - Readily available on any Linux host
  - Faster than libpcap
  - Will struggle at higher packet rates



DPDK

- Open-source framework for high speed packet processing
- Replaces NIC drivers with ones optimised for capture speed
    - Packets delivered directly to the user space process
    - Bypass kernel networking stack
- Can capture at 10Gbps line rate easily
- Reasonably mature project

DPDK
DATA PLANE DEVELOPMENT KIT

DPDK

- Only works on certain NICs
  - Interface is no longer visible to the kernel

- Setup process is not user-friendly
  - But getting better over time...

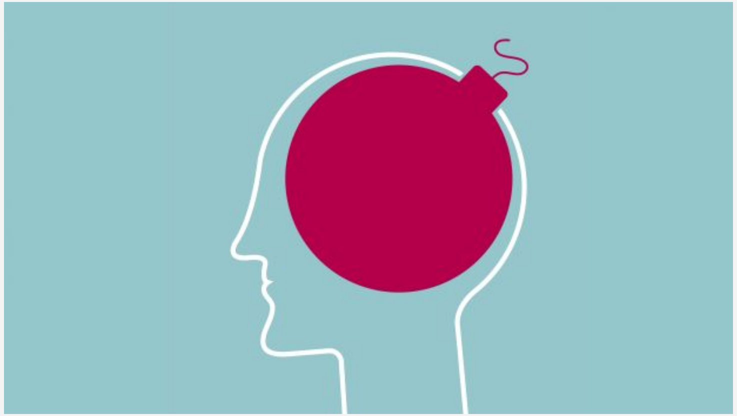
AF_XDP sockets

- Next generation raw socket designed for high speed capture
  - Linux 4.18 kernel onwards
  - Modern distros will have XDP enabled by default
  - Uses eBPF to intercept packets before the kernel stack

AF_XDP sockets

- Allows packets to be seen in user space with zero copy
  - Verified to handle 10Gbps line rate

- Less painful to use than DPDK



AF_XDP sockets

- Not all NIC drivers support XDP natively
  - Fall back to a slower software implementation

- Relatively new feature
  - Not many resources for non-experts

Other Options

- Endace DAG
  - Specialised hardware capture cards
  - Very high performance, accurate timestamping
  - Not cheap, individual cards no longer sold

- PF_RING
  - Kernel module implementing kernel bypass techniques
  - Not in mainline kernel, but very mature
  - Simple to use, supports most NICs

Libtrace

- Library for simplifying packet capture and analysis
    - Supports all of the mentioned methods with a single API
    - Hides much of the complexity
        - DPDK configuration is still a nuisance

- OpenLI uses libtrace to support each capture method
    - Choose whichever suits your needs best

How to Decide?

- AF_PACKET is perfect for smaller workloads
    - < 2Gbps is a rough guideline I would use

- In the past, I've recommended DPDK for high speed capture
    - Works very well, but learning curve is painful

- As XDP matures, this will become the recommended option
    - Easier to use, with great performance

Summary

- Packet capture
- Key factors that influence packet capture performance
  - Packet rate
  - Impact of the kernel networking stack
  - Taking advantage of multicore systems
- Different capture methods
  - Pros and cons

Next Time

- The components of OpenLI
    - Role of each component
    - Tips for planning their deployment
        - Hardware selection
    - Security considerations