

# Packet Capturing Challenges and Approaches

By

Paritosh Tripathi

C-DAC Mumbai

[paritosh@cdacmumbai.in](mailto:paritosh@cdacmumbai.in)

# Outline

- Introduction
- Background
- Approaches
- Challenges
- Path Ahead

# Introduction

- Need to monitor and troubleshoot network traffic
- Dimensions to packet capture: port spanning
- High speed networks: high data and high packet rates - Challenges
- 2 choices: commodity systems with associated software or customized hardware - Approaches

# Background

- Packet capture: to grab a copy of packets off of the wire before they are processed by the operating system.
- Packet arrives at network card: verifies checksum, extracts link layer data and triggers an interrupt
- The interrupt calls the corresponding kernel driver for packet capturing

# General overview

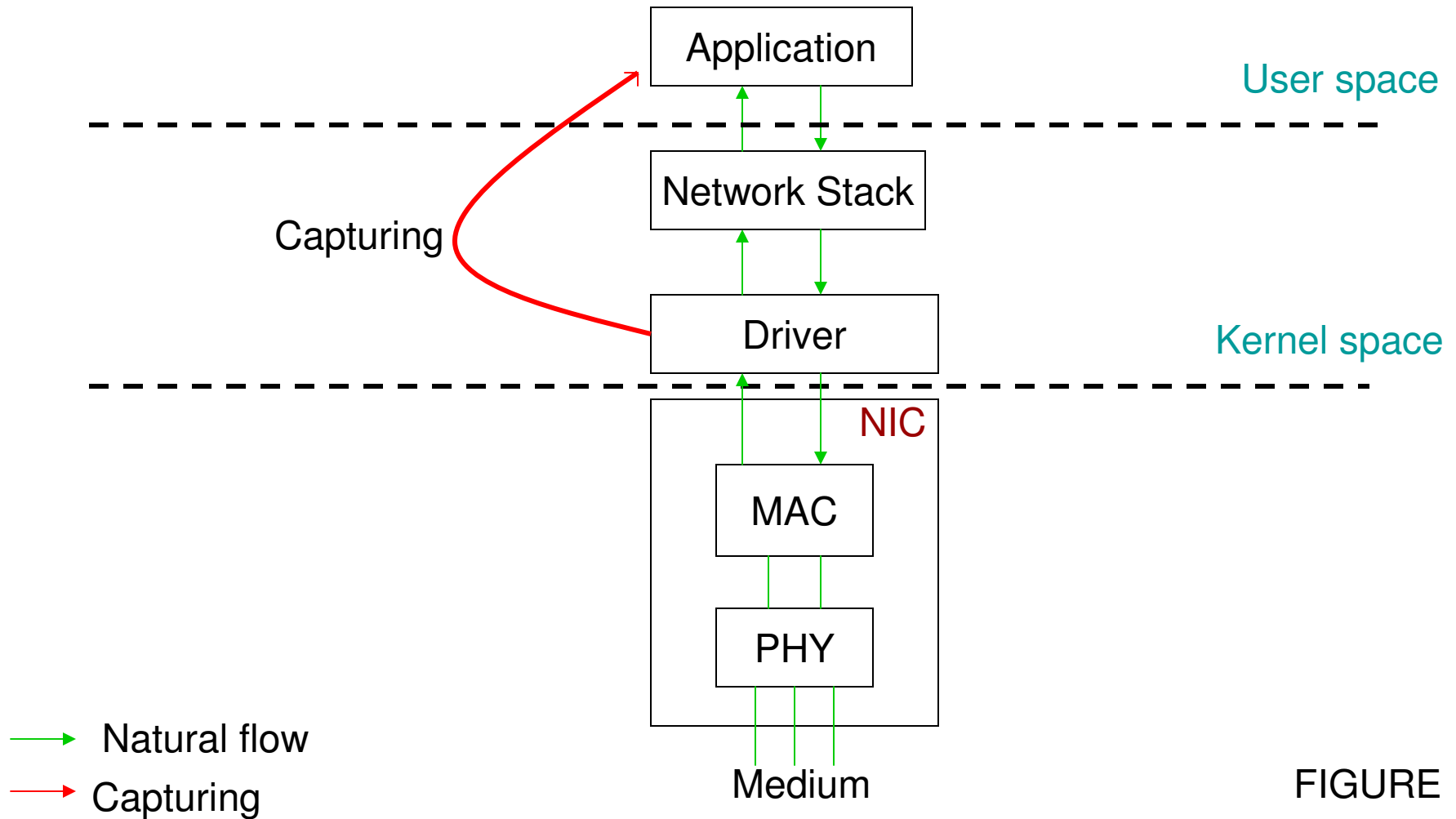


FIGURE 1

# Approaches

- 1) Under this approach: less than 1Gbps
- 2) Under this approach: between 1 and 10Gbps
- 3) Under this approach any network speed with any data rate

# Approach 1

- Libpcap is an open source library that provides a high level interface to network packet capture (figure 2).
- Step 1: find an interface to listen on
- Step 2: open the interface & set the byte count to be captured at a time
- Step 3: tell libpcap to start capturing packets

# Packet Capturing using Libpcap

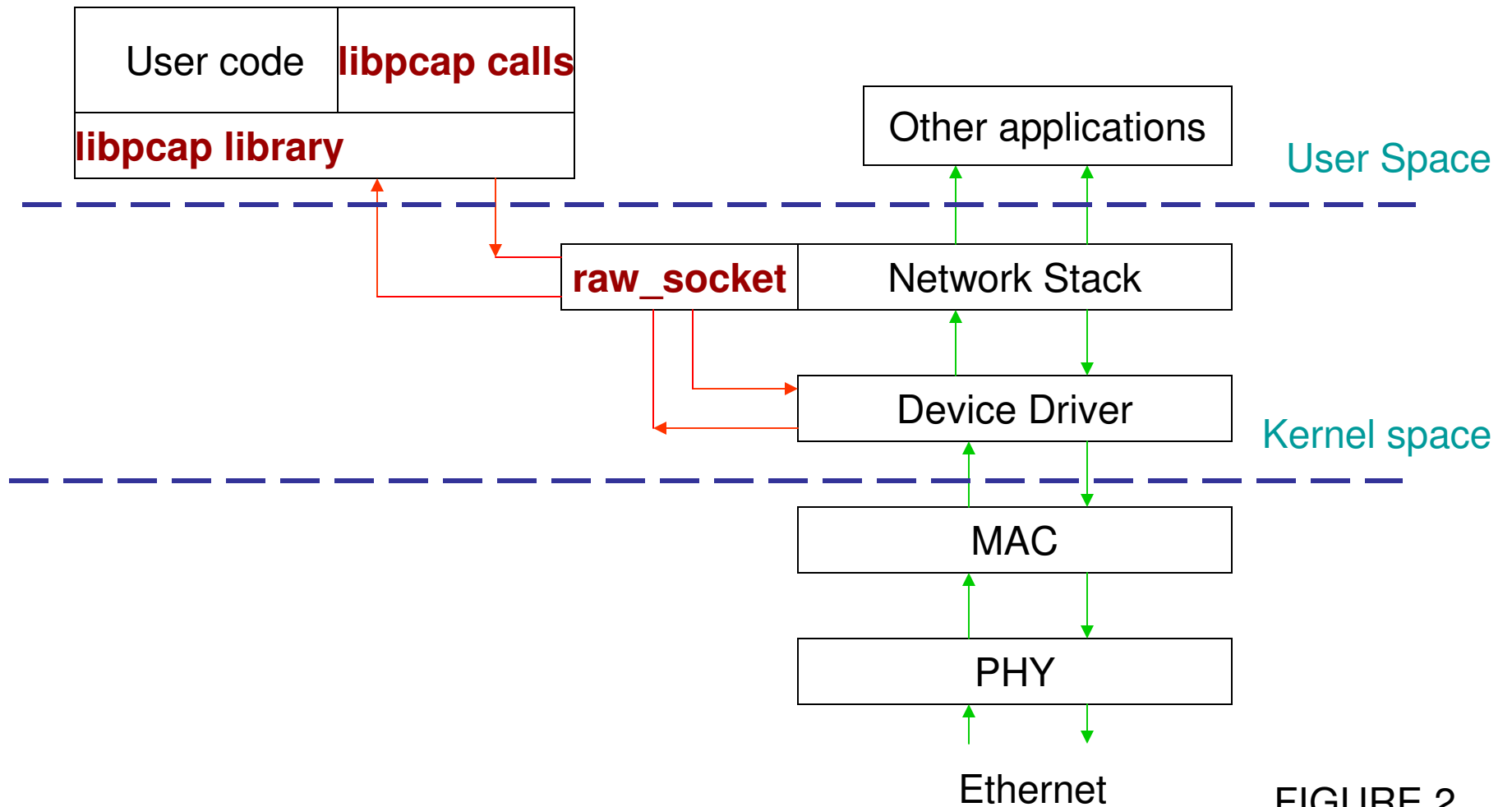


FIGURE 2

# mmaped version of libpcap

- Focus on moving packets from kernel space to user space
- Reduce per packet system calls
- Incoming packets are copied from the NIC to a kernel buffer that is shared between the kernel and the monitoring application

# The PF\_Ring Patch

- New type of socket optimized for packet capture based on circular buffer
- Packets are copied into the ring and not to the kernel network data structure
- Mmap primitives allows no overhead due to system calls as in case of socket calls
- Packets need not to be passed to upper layers

# Hang-on not all is captured!!!

- Network speeds are reaching (< 1 Gbps)
- The kernel is still involved in packet capturing process
- No way to exploit multiprocessing
- The processor which runs MAC stack on NIC is partially used
- Device drivers are not optimized for packet capture

# Approach 2: A way towards Gbps rates

- nCap library as a software solution
- Custom hardware: NIC card (interrupt coalescing) and a dedicated processor for packet capturing and analysis.
- Make common operations fast: Network processor usage

# Capturing using nCap

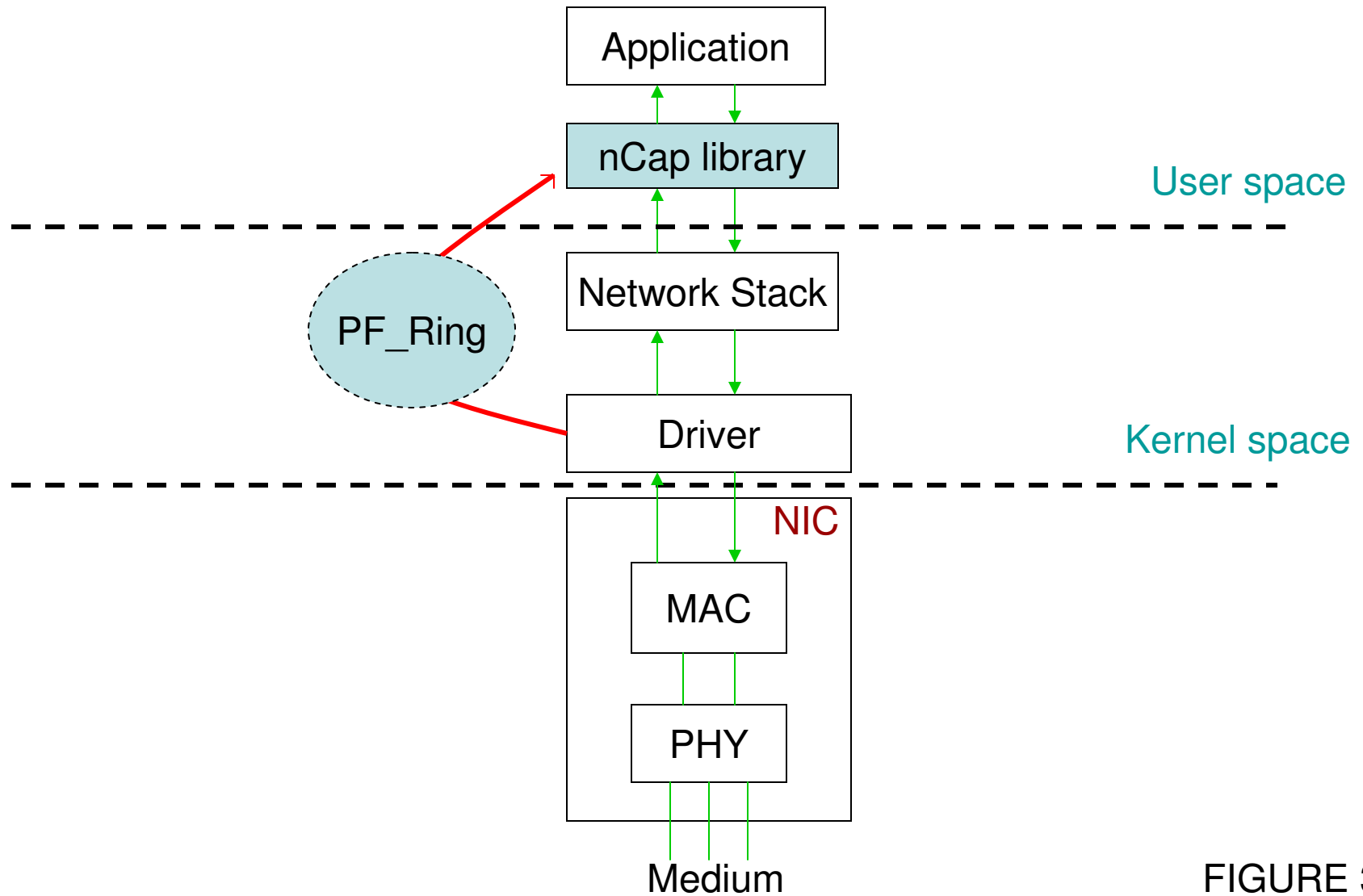
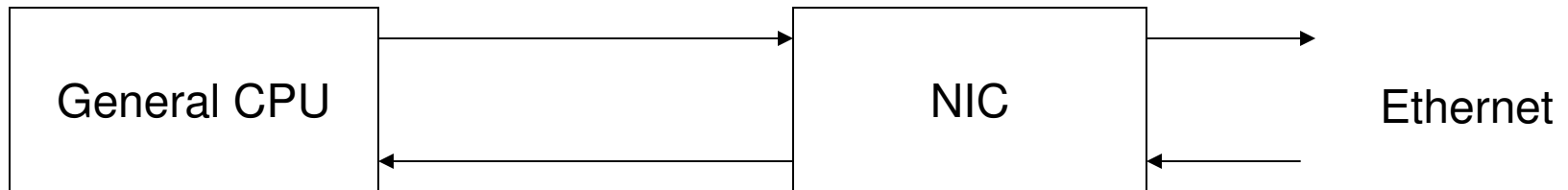


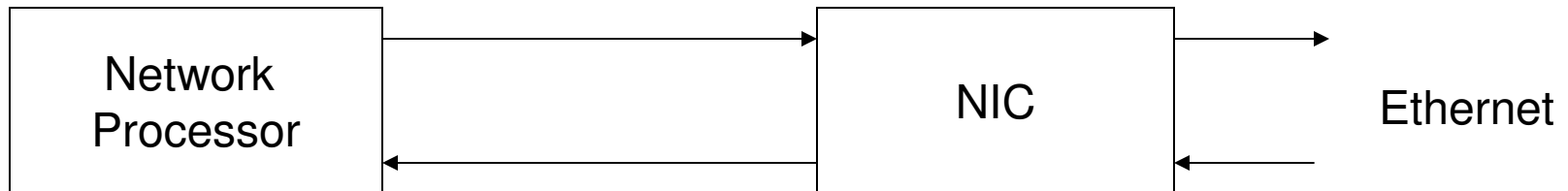
FIGURE 3

# Custom hardware



Runs only  
certain Application

Type 1



Certain  
Operations  
Fast

Type 2

FIGURE 4

# Approach 3: Specialized hardware (Net-FPGA is a case)

- Has gigabit ethernet ports
- Captures data and DMA writes to shared memory
- Host CPU is not involved in capturing
- CPU time saved can be used for packet processing
- Cheap, open-source and well supported

# Limitations with approach 2 & 3

- Cost implications are very high
- Limited usage

# Path Ahead

- Networks have already reached multi-gigabits of speed
- Special NICs with hardware accelerated packet forwarding and filtering (NIFIC)
- Endace DAG card, which uses multicore and complex load balancing algorithms