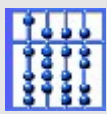


SEP

Packet Capturing Using the Linux Netfilter Framework

Ivan Pronchev
pronchev@in.tum.de



Today's Agenda

Goals of the Project

Motivation

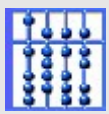
Revision

Design

Enhancements

tcpdump vs kernel sniffer

Interesting and Future Questions

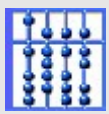


Goals of the Project

Approaching Linux netfilter framework

Developing kernel sniffer

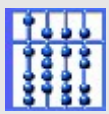
Comparing with an existing packet capturing tool



Motivation

Finding ways to improve capturing rates

Userspace vs Kernel space



Revision

Linux Netfilter Framework

Main Data Structures

Receive Livelock

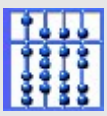
Processing Multiple Frames During an Interrupt(NAPI)

NAPI/non-NAPI Frame Reception

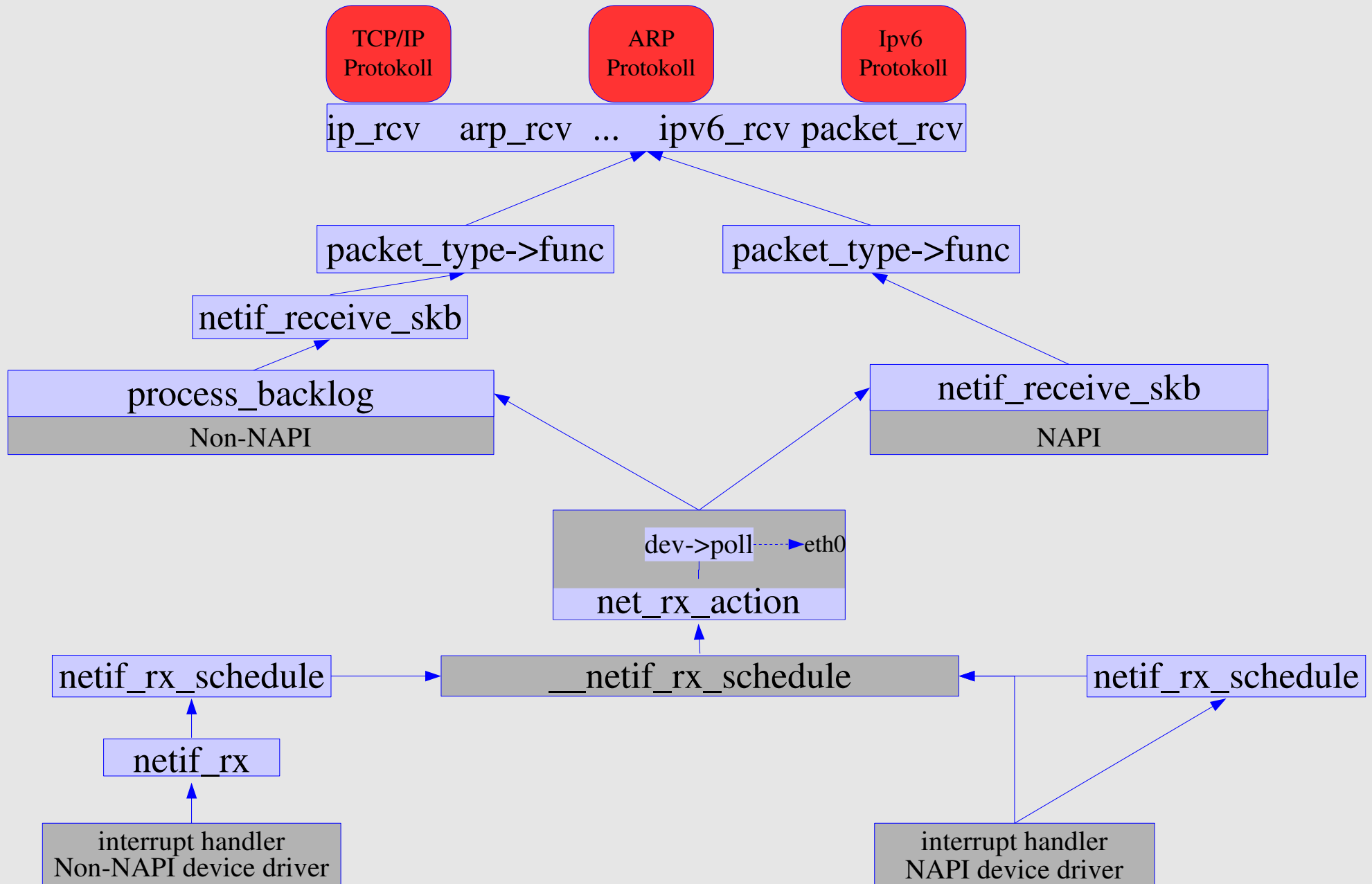
Packet Path through the IP Kernel Stack

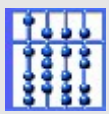
Netfilter Hooks in Details

Kernel Sniffer



NAPI/non-NAPI Frame Reception



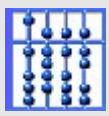


Design

How to capture packets ?

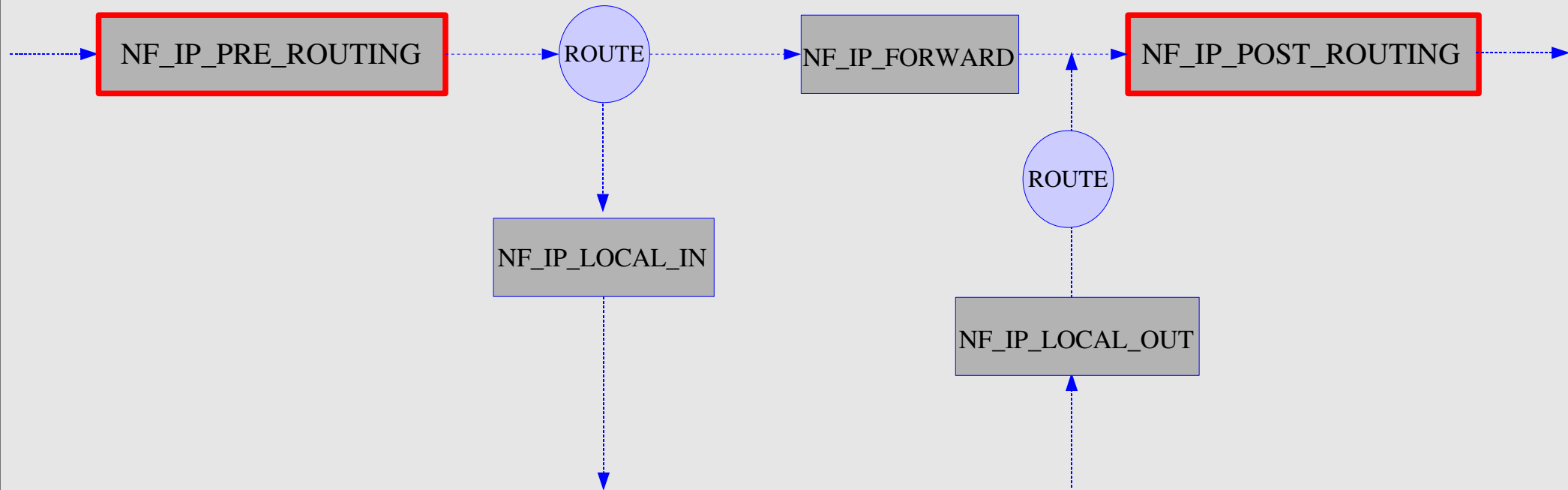
How file operations work in kernelspace ?

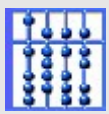
How to capture packets and write them into a file ?



Design

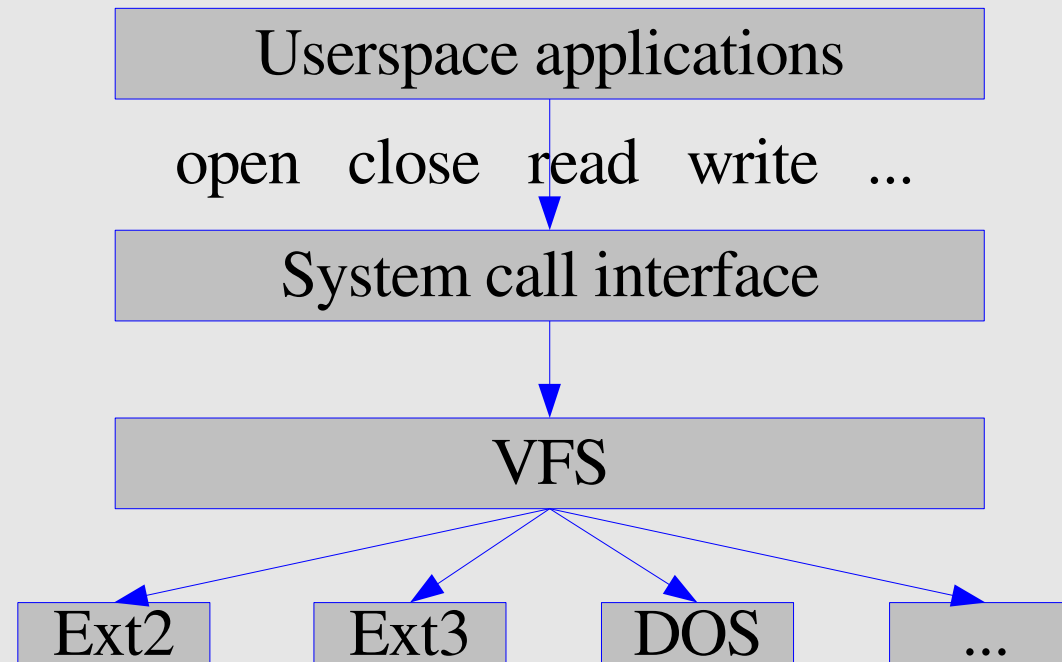
How to capture packets ?

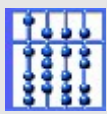




Design

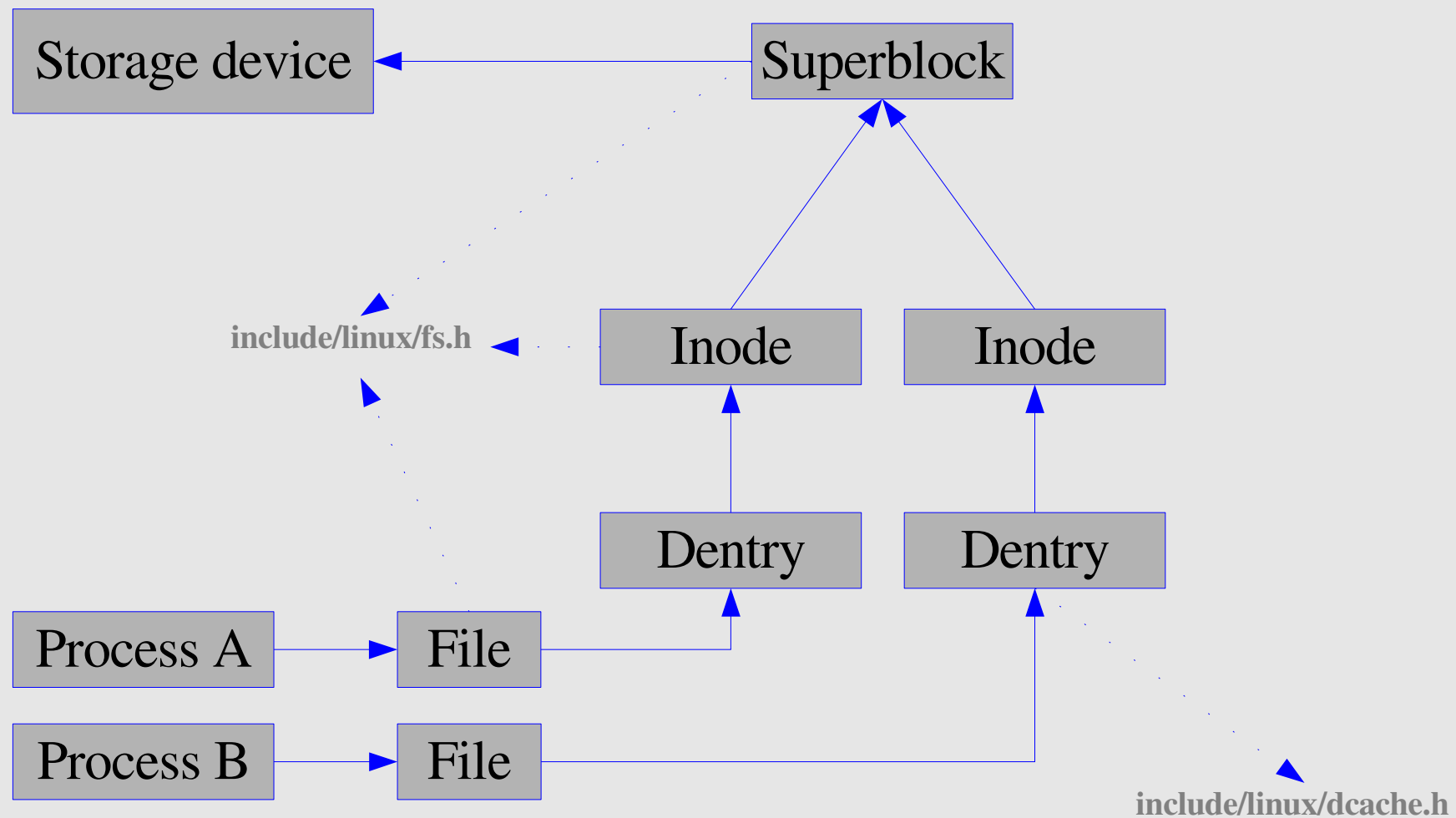
How file operations work in kernelspace ?

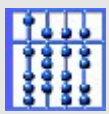




Design

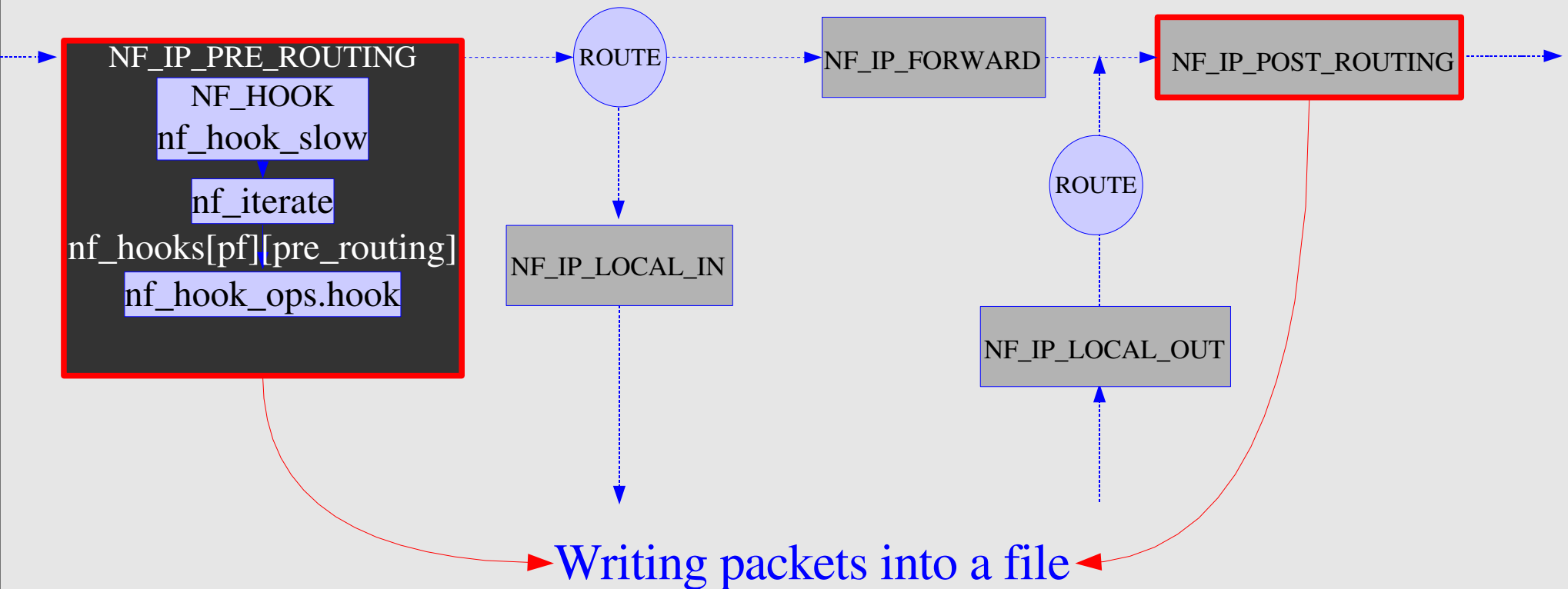
How file operations work in kernelspace ?



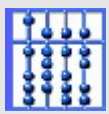


Design

How to capture packets and write them into a file ?

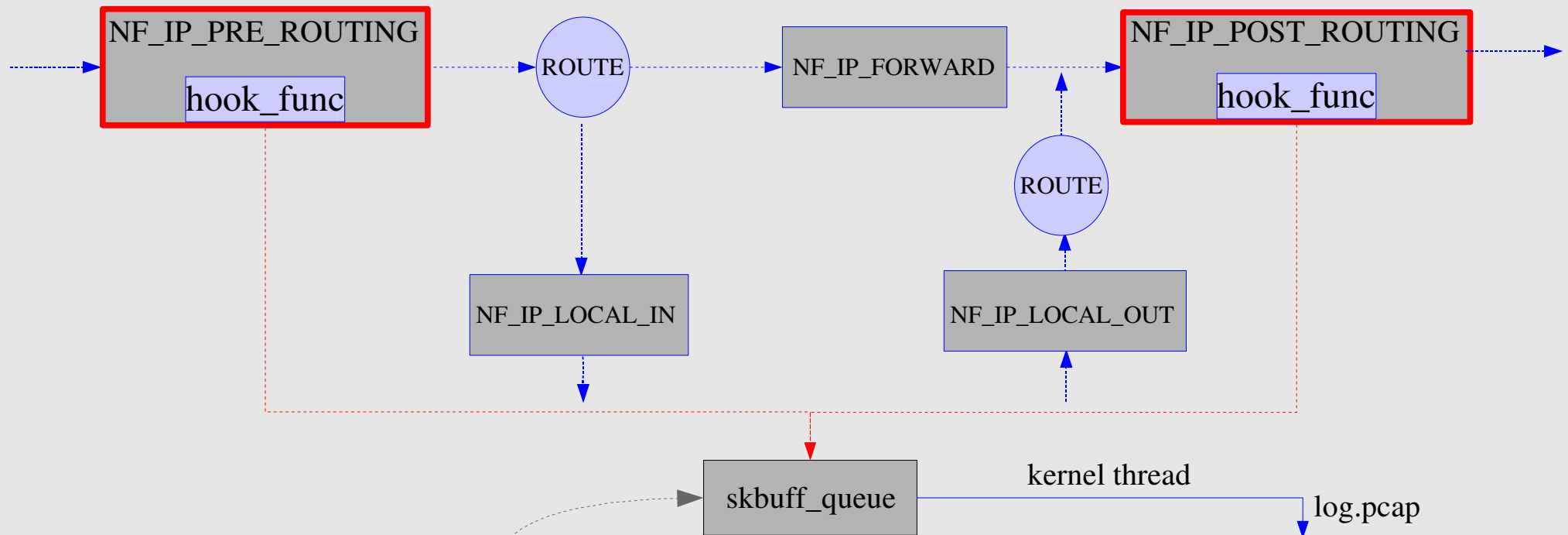


Not possible: context switch disabled in `nf_hook_slow` while writing invokes scheduling if necessary!

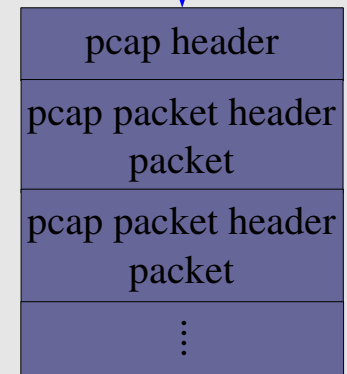


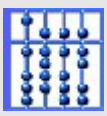
Design

How to capture packets and write them into a file ?

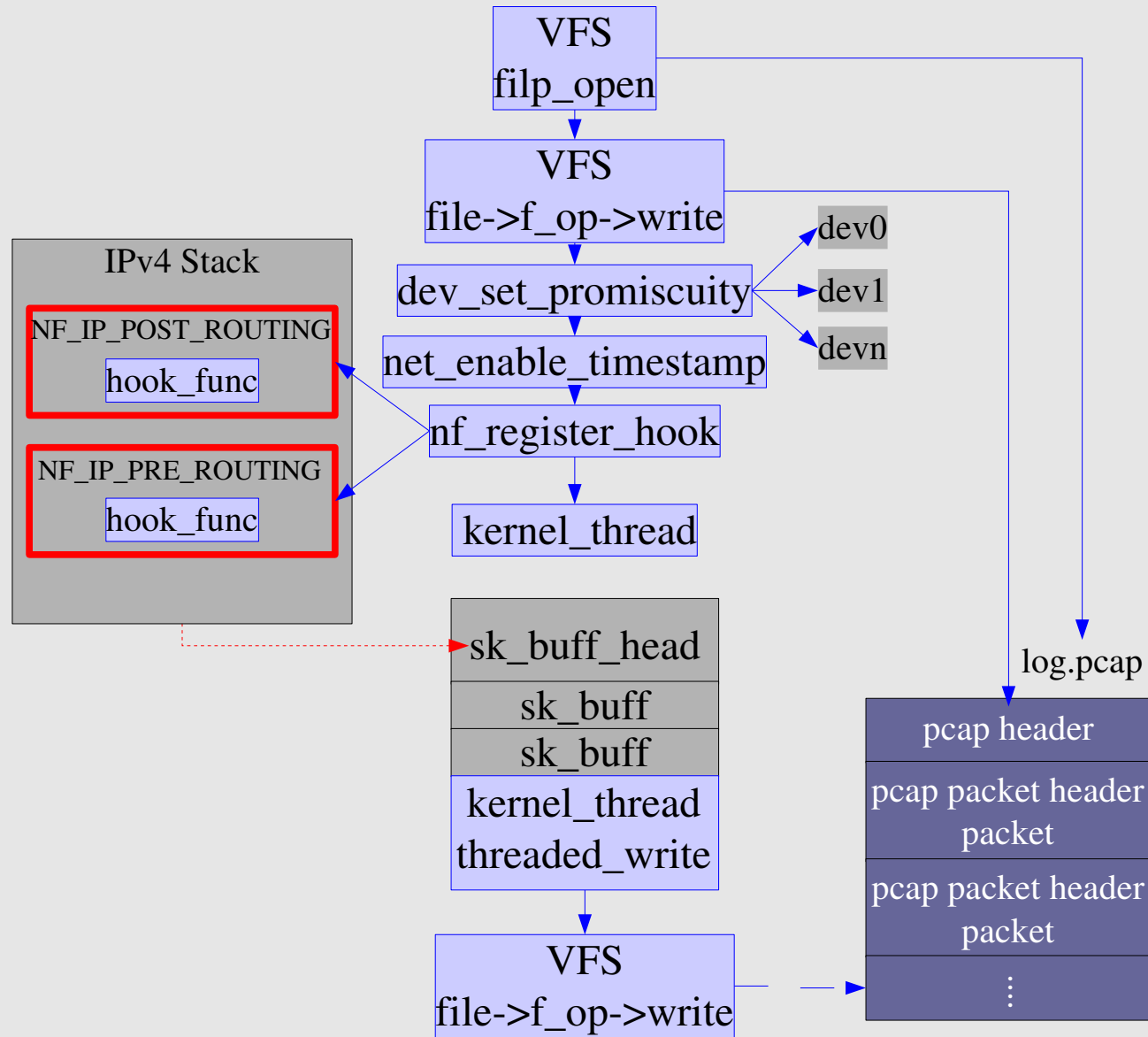


How to store the packets until further procession ?





Design





ip_rcv

```
int ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt, struct net_device *orig_dev)
{
```

1. When the interface is in promiscuous mode drop all the crap that it receives, do not try to analyze it.

```
    if (skb->pkt_type == PACKET_OTHERHOST)
        goto drop;
```

... ..

2. Call the prerouting netfilter hook.

```
    return NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, dev, NULL, ip_rcv_finish);
```

3. By error discard the **sk_buff** structure.

inhdr_error:

... ..

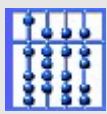
drop:

```
    kfree_skb(skb);
```

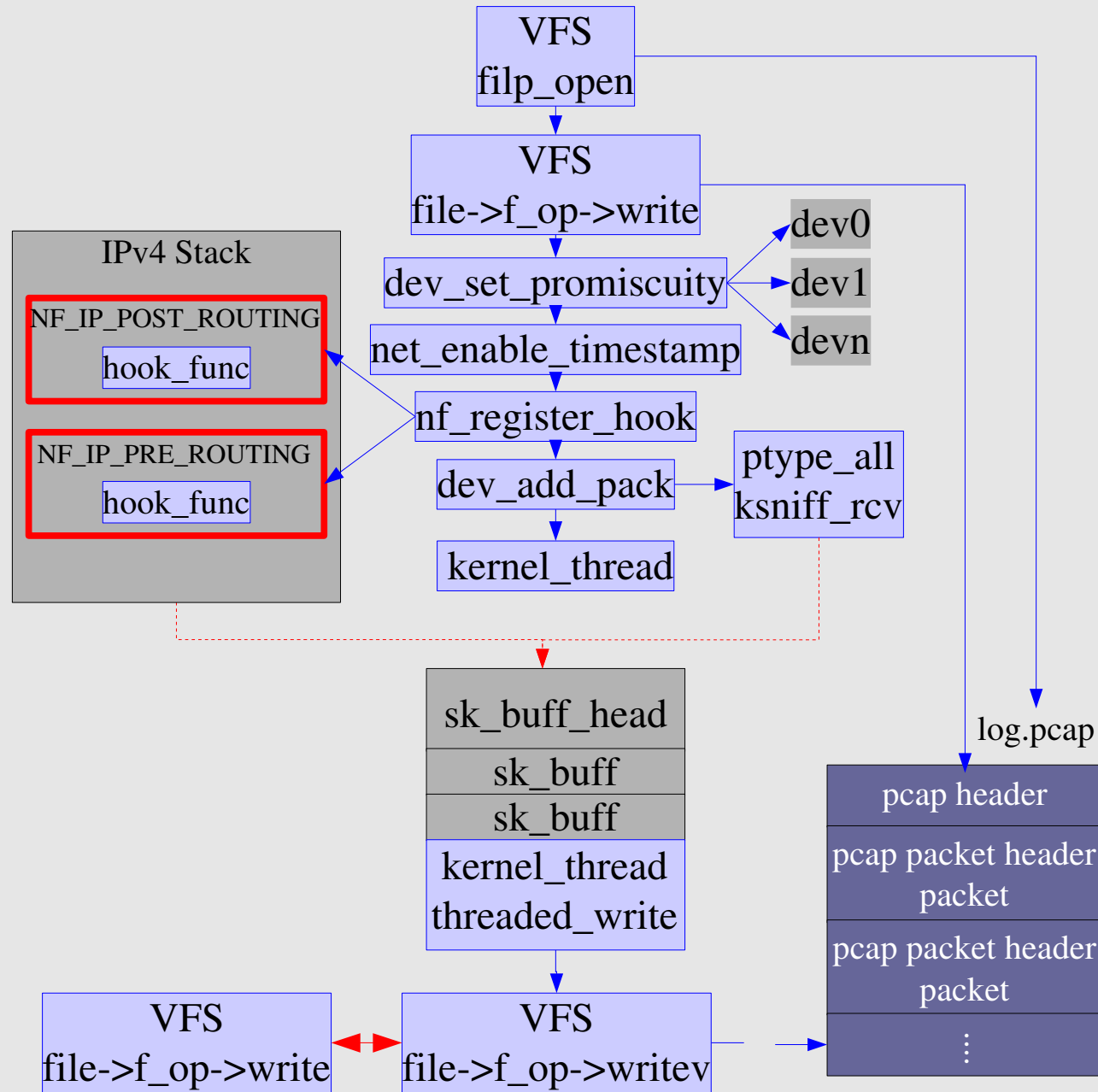
out:

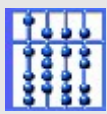
... ..

```
}
```



Design





Enhancements

Communication through the procs

- start,stop,restart

Interaction with the sniffer

- queue_size**

- device_name**

- logfile**

- snaplen**

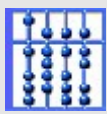
Statistics

- Errors

- Received packets

- Captured packets

Logging packets from a certain network device



tcpdump vs kernel sniffer

Test machine:

Athlon XP 1800, RAM:256

maximal disk's write speed ~ 34 MB/s

TEST 1 : kernel sniffer, snaplen=1500

Packets:2000000 (1496byte,0frags)
70808pps 847Mb/sec (847432454bps) errors: 0

Captured packets:603874

Received packets:655560

TEST 1: tcpdump, snaplen=1500

Packets:2000000 (1496byte,0frags)
70800pps 847Mb/sec (847344015bps) errors: 0

589831 packets captured

661719 packets received by filter



tcpdump vs kernel sniffer

TEST 2: kernel sniffer, snaplen=96

Packets:2000000 (1496byte,0frags)
70799pps 847Mb/sec (847331807bps) errors: 0

Captured packets:647783

Received packets:647783

TEST 2: tcpdump, snaplen=96

Packets:2000000 (1496byte,0frags)
70808pps 847Mb/sec (847431164bps) errors: 0

642799 packets captured

645014 packets received by filter

TEST 3: kernel sniffer, snaplen=1500

Packets:10.000.000 (1496byte,0frags)
47274pps 565Mb/sec (565784851bps) errors: 0

Captured packets:3791329

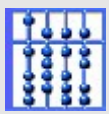
Received packets:9844006

TEST 3: tcpdump, snaplen=1500

Packets:10.000.000 (1496byte,0frags)
47088pps 563Mb/sec (563557308bps) errors: 0

3643704 packets captured

9930613 packets received by filter

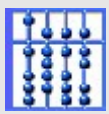


Interesting and Future Questions

Queue vs Ring-buffer

Direct IO vs non-Direct IO file operations

Finding ways to improve capturing rates



Thanks for the attention