Wire-speed Packet Capture and Transmission

Luca Deri <deri@ntop.org>

ntop.org

Packet Capture: Open Issues

- Monitoring low speed (100 Mbit) networks is already possible using commodity hardware and tools based on libpcap.
- Sometimes even at 100 Mbit there is some (severe) packet loss: we have to shift from thinking in term of speed to number of packets/second that can be captured analyzed.
- Problem statement: monitor high speed (1 Gbit and above) networks with common PCs (64 bit/66 Mhz PCI/X/Express bus) without the need to purchase custom capture cards or measurement boxes.

Libpcap Performance [1/2]

Packet Size (Bytes)	Speed (Mbit)	Speed (Pkt/sec)	Linux 2.6.1 with NAPI and standard libpcap	Linux 2.6.1 with NAPI and mmap()	FreeBSD 4.8 with Polling
64	90	175’000	2.5%	14.9%	97.3%
512	710	131’000	1.1%	11.7%	47.3%
1500	836	70’000	34.3%	93.5%	56.1%

Percentage of captured packets

Testbed:

- Sender: Dual 1.8 GHz Athlon, Intel GE 32-bit Ethernet card
- Collector: Pentium III 550 MHz, Intel GE 32-bit Ethernet card
- Traffic Generator: stream.c (DoS)

Libpcap Performance [2/2]

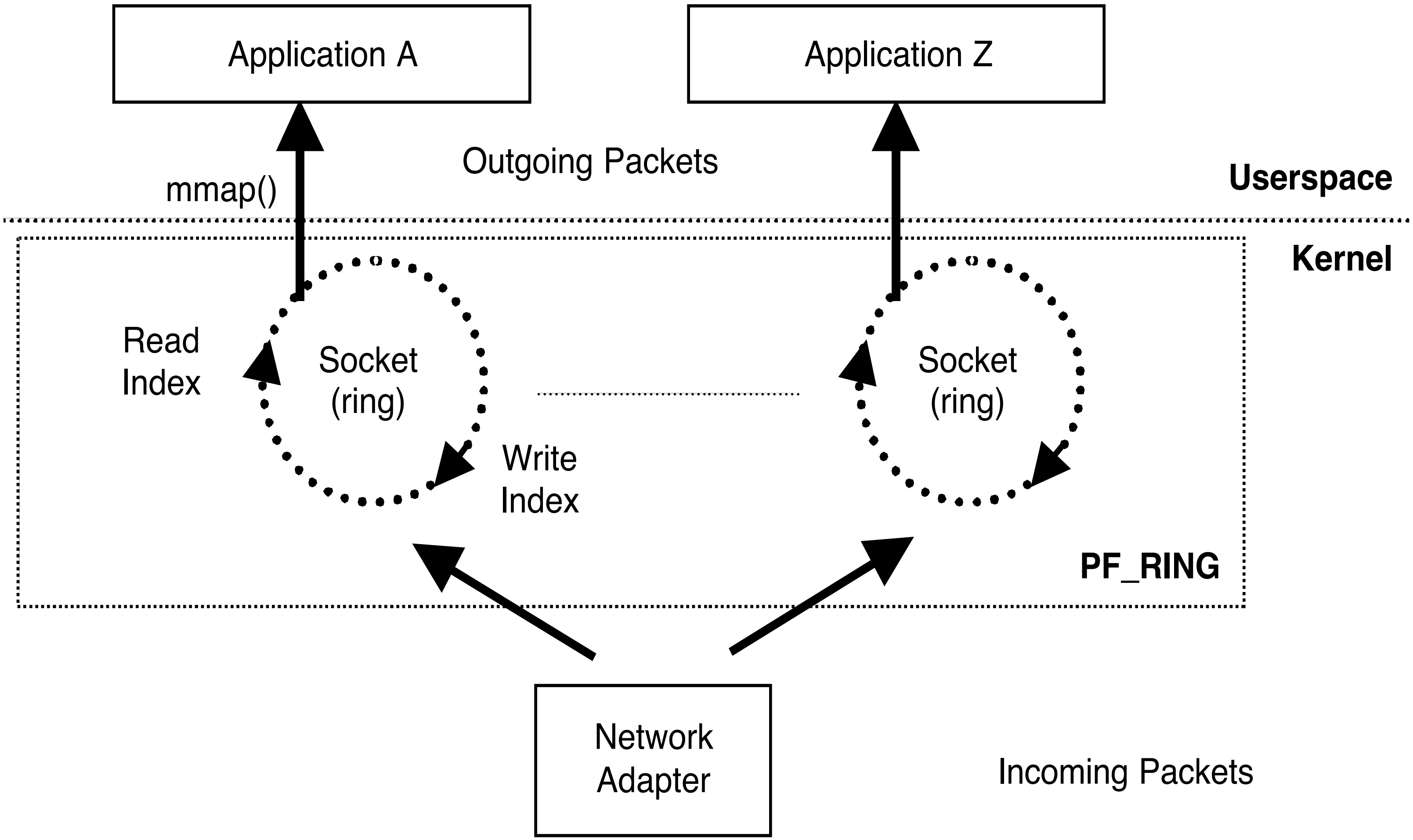
Using mmap() for direct packet access into the kernel has:

- significantly improved the capture performance
- partially solved the problem as Linux is still quite slow. This is somehow a demonstration that context switching (kernel to userland) is an issue but it's not the real issue that slows down the capture process.

Further comments:

- Device Polling significantly improved the performance on a 100 Mbit Ethernet card
- Linux still performs much worse than FreeBSD at userspace
- Linux kernel performance is basically the same of FreeBSD at userspace

Proposed Solution:
Socket Packet Ring (PF_RING)

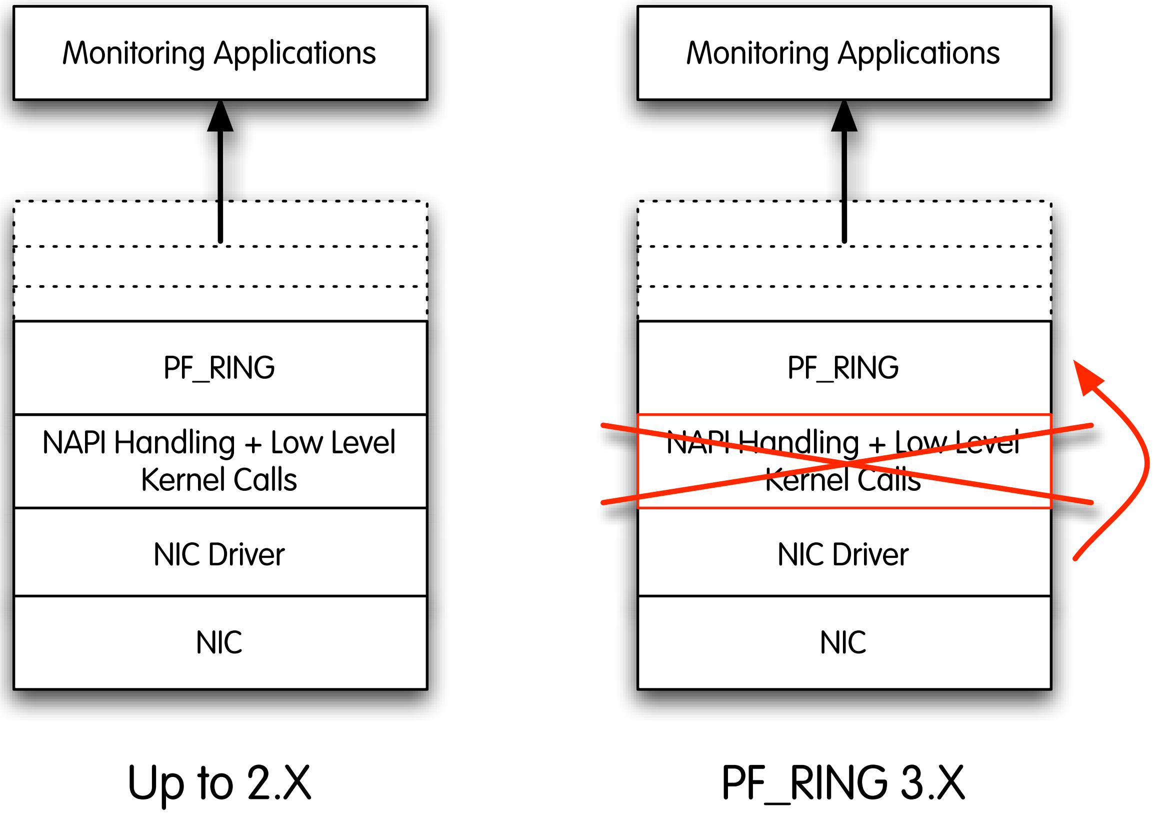


ntop.org

PF_RING Features

- Linux kernel patch (2.4.x and 2.6.x) for high-speed packet capture.
- In a nutshell it reduces the packets journey from the NIC to the user applications.
- It adds a new type of socket (PF_RING) that can be used by existing (PF_PACKET) applications.
- The (legacy) libpcap library has been extended in order to support PF_RING.

PF_RING 3.x: Speed



- Advantage: Major speed bump.
- Limitation: NIC driver needs (very minor) modifications.

ntop.org

PF_RING 3.x: Evaluation [1/2]

Evaluation:

- Major improvement with respect to Linux with NAPI
- It can exploit both polling and Linux 2.4.x/2.6.x
- Users (stillsecure.com) sell accelerated Snort/PF_RING able to run at 1.6/1.8 Gbit (aggregate) on fast Opteron PCs

Open Issues

- Packet loss: still some packets are lost on Linux.
- CPU Usage: on Linux there still some packet loss although the CPU usage is very low (< 30% on Linux, > 98% on FreeBSD).

PF_RING 3.x: Evaluation [2/2]

Packet Size (Bytes)	Linux 2.4.23 with NAPI, RT_IRQ and Ring (Pkt Capture)	Linux 2.4.23 with NAPI, RT_IRQ and Ring (nProbe)
64	550'789 [~202 Mbit]	376'453 [~144 Mbit]
512	213'548 [~850 Mbit]	213'548 [~850 Mbit]
1500	81'616 [~970 Mbit]	81'616 [~970 Mbit]

Captured Packets and nProbe Flow Generation (packet/sec)

Testbed:
Sender: Dual 1.8 GHz Athlon, Intel GE 32-bit Ethernet card
Collector: Pentium 4 1.7 GHz, Intel GE 32-bit Ethernet card
Traffic Generator: stream.c (DoS)

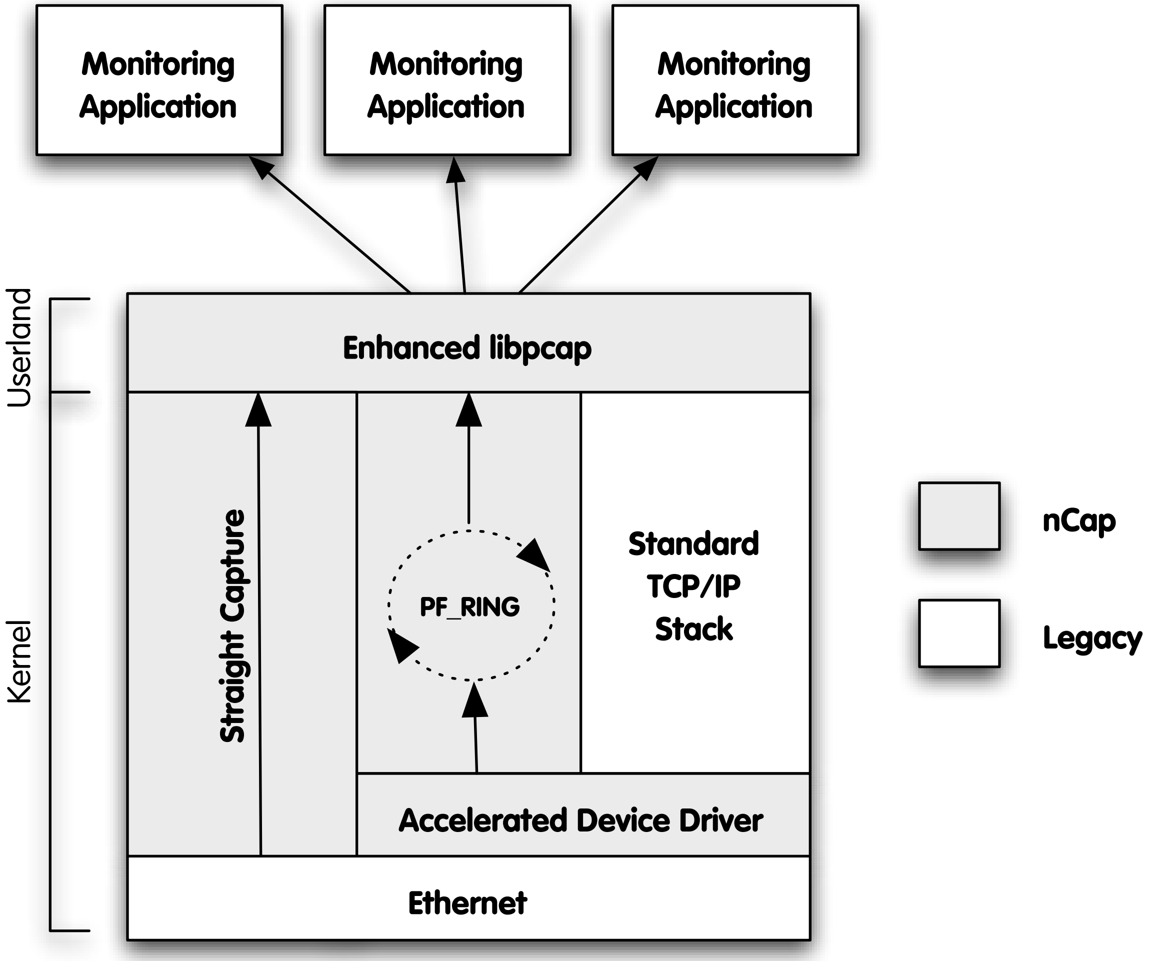
PF_RING: Open Issues

- The kernel is still involved in the capture process (overhead).
- Kernel packet polling is implemented only on the first CPU (no way to really exploit multiprocessing).
- Fetching full packets is costly as it requires extra kernel work (memcpy).
- The NPU on the ethernet card is partially used as most of the processing is done on the main CPU.
- Device drivers are not optimized for packet capture: too many memory allocations/copy/free.

What’s next?

- Completely remove the kernel from the packet capture process.
- Avoid packet copy at all.
- Fully exploit the NPU that’s on the ethernet card.
- Use the main CPU(s) for packet processing and for fetching packets from network adapters.
- Rethink network device drivers and optimize them for packet capture.

Welcome to nCap



ntop.org

nCap Features

	Packet Capture Acceleration	Wire Speed Packet Capture	Number of Applications per Adapter
Standard TCP/IP Stack with accelerated driver	Limited	No	Unlimited
PF_RING with accelerated driver	Great	Almost	Unlimited
Straight Capture	Extreme	Yes	One

nCap Internals

- nCap maps at userland the card registers and memory.
- The card is accessed by means of a device /dev/ncap/ethX
- If the device is closed it behaves as a "normal" NIC.
- When the device is open, it is completely controlled by userland the application.
- A packet is sent by copying it to the TX ring.
- A packet is received by reading it from the RX ring.
- Interrupts are disabled unless the userland application wait for packets (poll()).
- On NIC packet filtering (MAC Address/VLAN only).

nCap Evaluation

- It currently supports Intel 1 GE copper/fiber cards.
- GE Wire speed (1.48 Mpps) full packet capture starting from P4 HT 3 GHz.
- Better results (multiple NICs on the same PC) can be achieved using Opteron machines (HyperTransport makes the difference).
- The nCap speed is limited by the speed applications fetch packets from the NIC, and the PCI bus.

nCap Comparison (1 Gbit)

	Maximum Packet Loss at Wire Speed	Estimated Card Price	Manufacturer
DAG	0 %	> 5-7 K Euro	Endace.com
nCap	0.8 %	100 Euro	
Combo 6 (Xilinx)	5 %	> 7-10 K Euro	Liberouter.com

Source Cesnet (http://luca.ntop.org/ncap-evaluation.pdf)

Further nCap Features

- High-speed traffic generation: cheap trafgen as fast as a hardware trafgen (>> 25’000 Euro)
- Precise packet generation.
- Precise packet timestamping on transmission (no kernel interaction): suitable for precise active monitoring.
- Enhanced driver currently supports Intel cards (1 Gb Ethernet).
- Support of PCI Express cards.

Availability

- Paper and Documentation:
  http://luca.ntop.org/

- PF_RING
  http://www.ntop.org/PF_RING.html

- nCap Live CD:
  http://luca.ntop.org/nCap/