Maximizing Packet Loss Monitoring Accuracy for Reliable Trace Collections

Javier Rubio-Loyola, Dolors Sala, Ali Ismail Ali

Presented by: Dr. Javier Rubio-Loyola
Universitat Pompeu Fabra, Barcelona

16th IEEE LANMAN: September 3-6, Cluj-Napoca, Romania

Motivation

Traffic traces:

- tool to design and analyze telecom infrastructures
- their accuracy and reliability have impact on the quality of network estimations, modeling, analysis, design and evaluation of network protocols, etc.

Traffic collection set ups with significant packet drops leave incomplete trace files of network traffic

The analysis over such traces may be incomplete unless data losses are explicitly reported to be considered also as part of the analysis

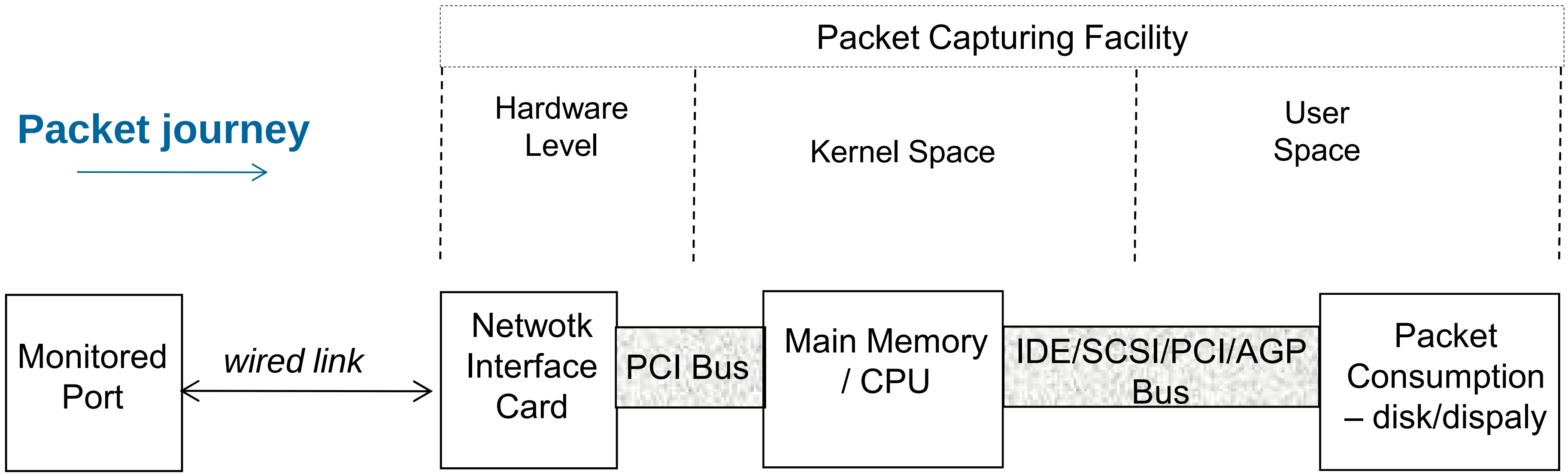
This work targets the maximization of packet loss monitoring accuracy for reliable packet trace captures

Background

Libpcap is a common system-independent interface for user-level packet capture

Tcpdump is by far the most famous program using libpcap for packet capture (this work elaborates on commodity soft- and hardware)

The research community has witnessed libpcap/tcpdump performance limitations with speeds of 1Gbps and higher[1]



Packet Capturing Facility

Packet journey

Hardware Level

Kernel Space

User Space

Monitored Port

wired link

Netwotk Interface Card

PCI Bus

Main Memory / CPU

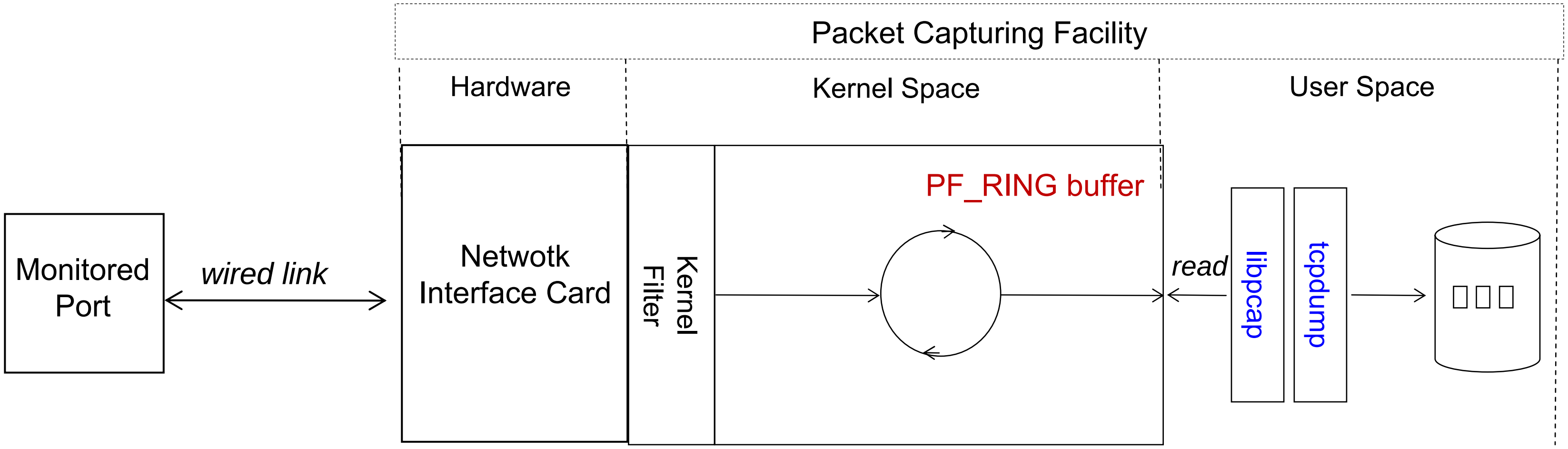
IDE/SCSI/PCI/AGP Bus

Packet Consumption – disk/dispaly

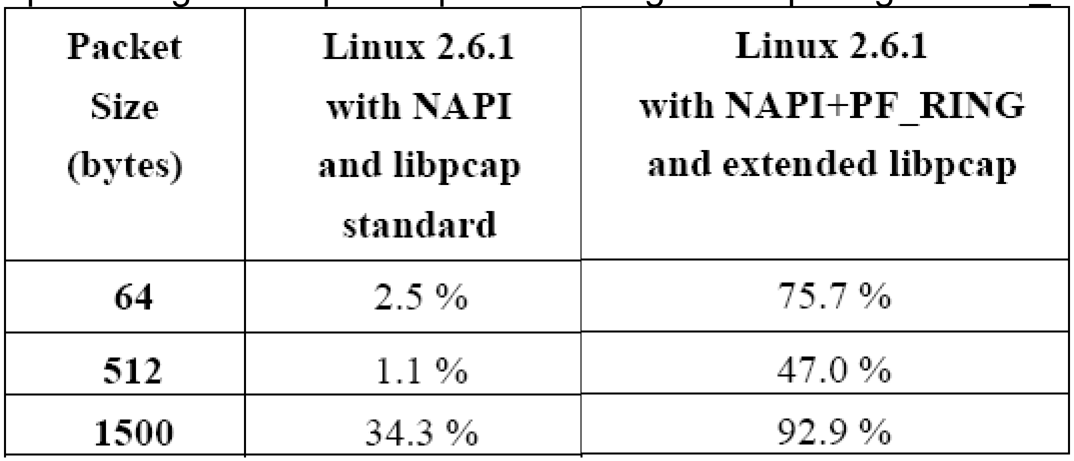
[1]Luca Deri. High-Speed Dynamic Packet Filtering. Journal of Network and System Management, June 2007

Background

PF_RING kernel buffer[2] drastically enhances the performance of packet capturing facilities from a kernel perspective



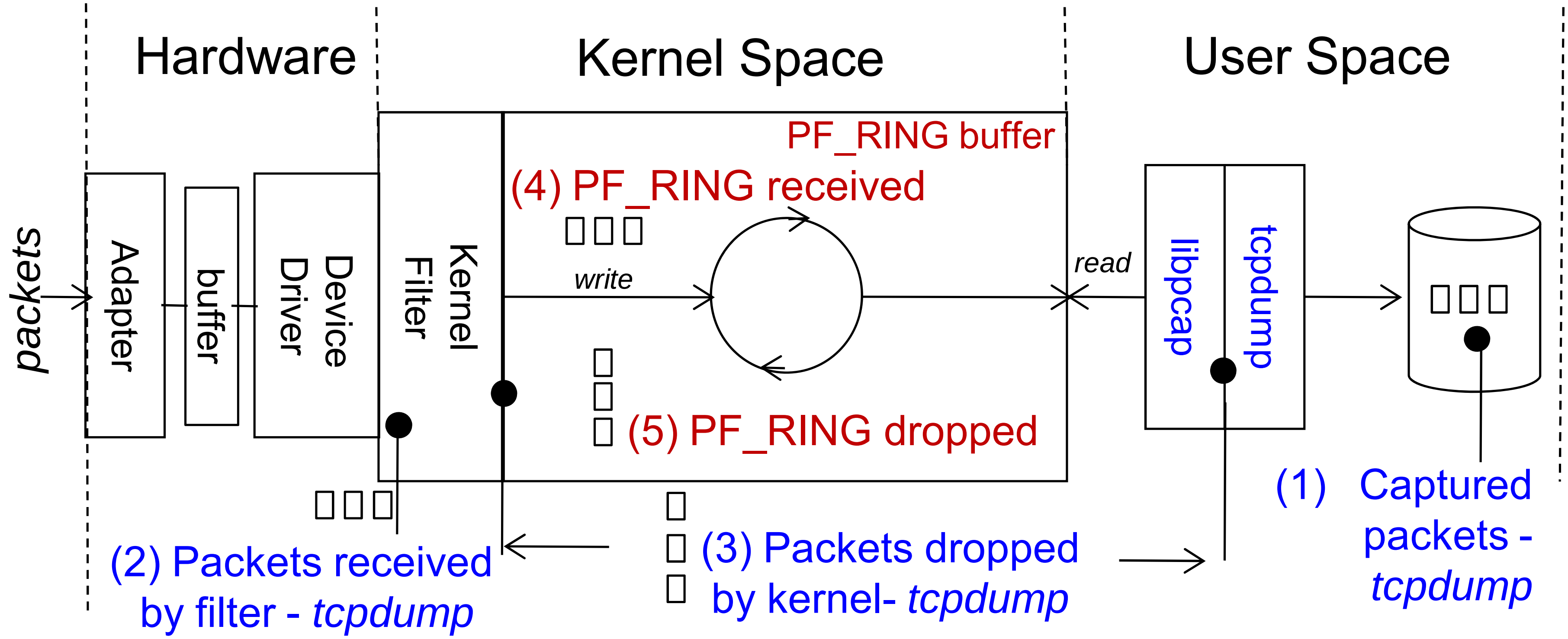
Some percentages of captured packets using kernel polling with PF_RING[2]



Packet Size (bytes)	Linux 2.6.1 with NAPI and libpcap standard	Linux 2.6.1 with NAPI+PF_RING and extended libpcap
64	2.5 %	75.7 %
512	1.1 %	47.0 %
1500	34.3 %	92.9 %

[2] Deri L: Improving Passive Packet Capture: Beyond Device Polling. Intl. Sys. Admin. & Net Eng , 2004

Available metrics for packet loss monitoring



The packet loss computation:

Packets received by filter (2) – captured packets (1) covers the wider area of the packet journey

Sources of unreliable or inaccurate packet loss computations

- Losses are likely to occur at:
  - hardware level
  - connection between hardware and the Operating System (kernel space)[3]

- Not measured, reported and/or associated to any of the earlier described packet-aware indicators

The above leads to potentially provide unreliable packet traces due to unreported or inaccurate packet loss monitoring

[3] Degioanni, L.et al. Profiling and optimization of software-based network-analysis applications
15th Symposium on Computer Architecture and High Performance Computing, 2003

Approach for Accurate Packet Loss Monitoring

- Meta-data and meta-trace collection
  - *"relevant information"* (metadata) to enable the computation of the difference between the generated packets and the captured ones
  - *meta-trace* – a collection of metadata – can be used to determine the reliability of the packet trace collections (post-processing)

- Integral framework for accurate packet loss monitoring
  - Absolute packet losses: from hardware level to user space
  - Rely on IEEE standard information
  - meta-trace collection in parallel with packet traces

- Practical requirements: non-instrusive and costly affordable

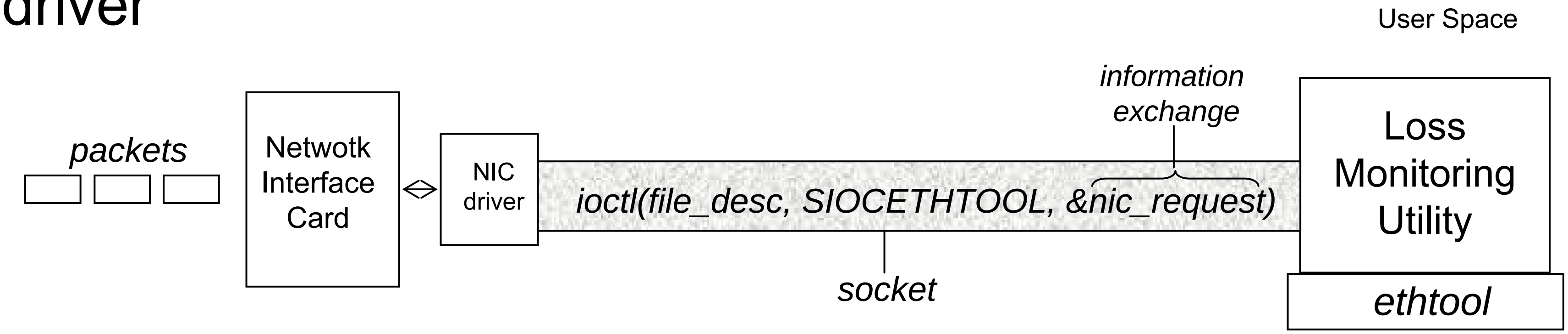
[3] *R. Loyola et al "Using Linear Temporal Model Checking for Goal Oriented Policy Refinement Frameworks". POLICY 2005*

Practicality of the approach

Hardware statistics retrieval from user space

- A two step process:

1. establish communication with the capturing interface’s NIC driver



- ioctl calls with a file descriptor (standard datagram socket), a device-dependent request code and a pointer to memory (step 2)

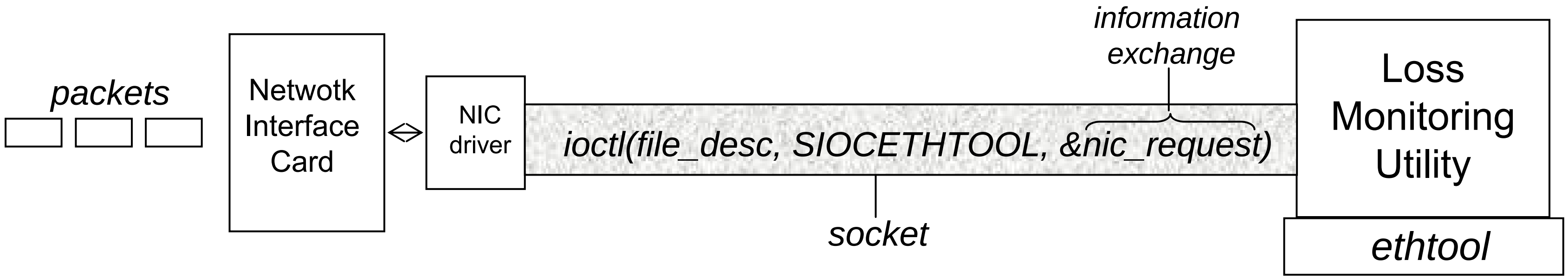
- a unique ioctl command named “SIOCETHTOOL” is issued by our Loss Monitoring Utility so that the NIC driver **always** returns ethtool-like data

Practicality of the approach

Hardware statistics retrieval from user space

- A two step process:

2. information exchange with NIC drivers



- The pointer to memory (nic_request) is filled with the command to retrieve NIC stats specified as a data structure.
- Whenever an ioctl call is issued the NIC stats are stored in the above data structure.

Once in the data array structure, the NIC statistics can be casted into numerical values (long long)

Practicality of the approach

Metadata and meta-trace collection

- Ideal absolute packet loss computation:

|PktLoss|= ∑ (receiveOK + frameTooLong + frameCheckError + lengthError + alignmentError + MAC rx errors) – tcpdump captured packets … (i)

- Advantage: it is based on IEEE 802.3 standard indicators[4]
- Drawback: Some relevant indicators are MANDATORY, however some are optional and other are recommended (depend on the NIC manufacturer)

Maximization of the packet loss monitoring is a function of the available indicators in (i) for the actual NIC

[4] *IEEE Std 802.3™-2005. IEEE Standard for Information technology*

Practicality of the approach

Metadata and meta-trace collection

- Example, for a general purpose card:

Status report	IEEE Rx Standard counter	Condition	D-Link DGE-530T NIC
receiveOK	aFramewReceivedOK	Mandatory	rx_unicast, rx_broadcast, rx_multicast
frameCheckError	aFrameCheckSequenceErrors	Mandatory	rx_fcs_error
alignmentError	aAlignmentErrors	Mandatory	rx_jabber
frameTooLong	aFrameTooLongErrors	Optional	rx_too_long
lengthError	aInRangeLengthErrors	Optional	rx_runt
	aOutOfRangeLenghtField	Optional	Not available (included in rx_too_long)
MAC sub layer errors	aFramesLostDueToIntMACRcvError (rx buffer overflow, rx abort from MAC, rx fifo overflow)	Recommended	fifo_overflow (only)

The loss monitoring utility should produce the meta-trace:

cpu_time, rx_broadcast, rx_multicast, rx_unicast, fifo_overflow, rx_jabber, rx_runt, rx_too_long, rx_fcs_error

[4] *IEEE Std 802.3™-2005. IEEE Standard for Information technology*

Practicality of the approach

Post-processing

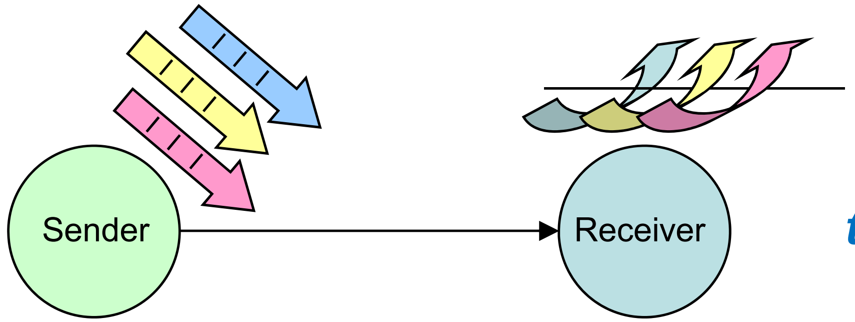
- Post-processing step to compute the absolute packet loss data
  - Inputs: Meta-trace MT1 and tcpdump packet traces
- For the above example post-processing produces the following meta-trace (MT2):

cpu_time, rx_broadcast, rx_multicast, rx_unicast, fifo_overflow, rx_jabber, rx_runt, rx_too_long, rx_fcs_error, tcpdump_captured_packets, absolute_packet_loss, pkt_loss_percentage ..... (MT2)

[4] IEEE Std 802.3™-2005. IEEE Standard for Information technology

Performance Evaluation

Test-bed and measurement points



Sender

Receiver

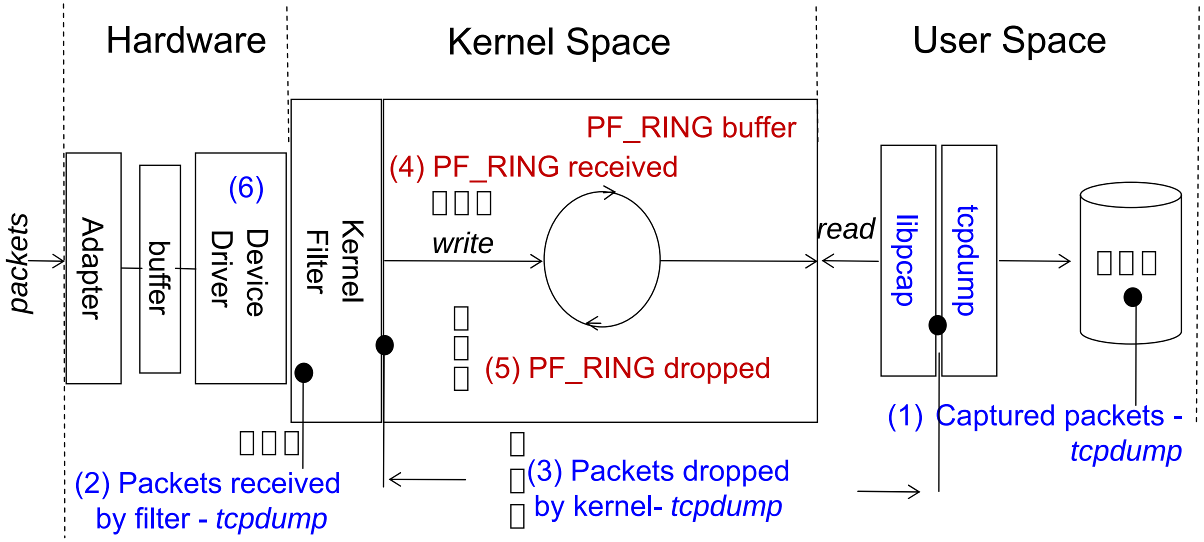
test-bed

Intel(R) Pentium(R) 4 CPU 3.00GHz
RAM: 1G
D-Link Gigabit Ethernet Adapter
(or) Marvell 88E8053 GbE

Intel(R) Pentium(R) 4 CPU 3.00GHz
RAM: 1G
D-Link DGE-530T Gigabit Ethernet Adapter
(or) Realtek RTL-8169 Gigabit Ethernet Adapter
Samsung Hards Disk SATA, 7200 rpm 140GB

Measurement Points

Collection set to 1sec



Hardware

Kernel Space

User Space

packets

Adapter

buffer

Device Driver

(6)

Kernel Filter

PF_RING buffer

(4) PF_RING received

write

(5) PF_RING dropped

read

libpcap

tcpdump

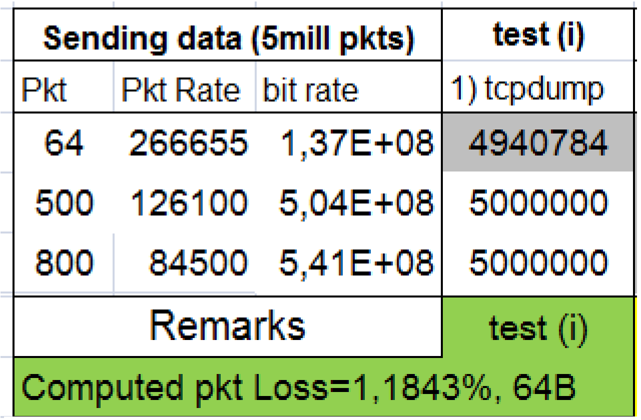
(1) Captured packets - *tcpdump*

(2) Packets received by filter - *tcpdump*

(3) Packets dropped by kernel- *tcpdump*

Performance Evaluation

Overhead tests



Sending data (5mill pkts)			test (i)
Pkt	Pkt Rate	bit rate	1) tcpdump
64	266655	1,37E+08	4940784
500	126100	5,04E+08	5000000
800	84500	5,41E+08	5000000
Remarks			test (i)
Computed pkt Loss=1,1843%, 64B			

Tests avoiding disk overhead:

Test (i) tcpdump capturing without loss monitoring
Test (ii) tcpdump capturing with PF_RING-aware loss monitoring
Test (iii) tcpdump capturing with our system wrapper

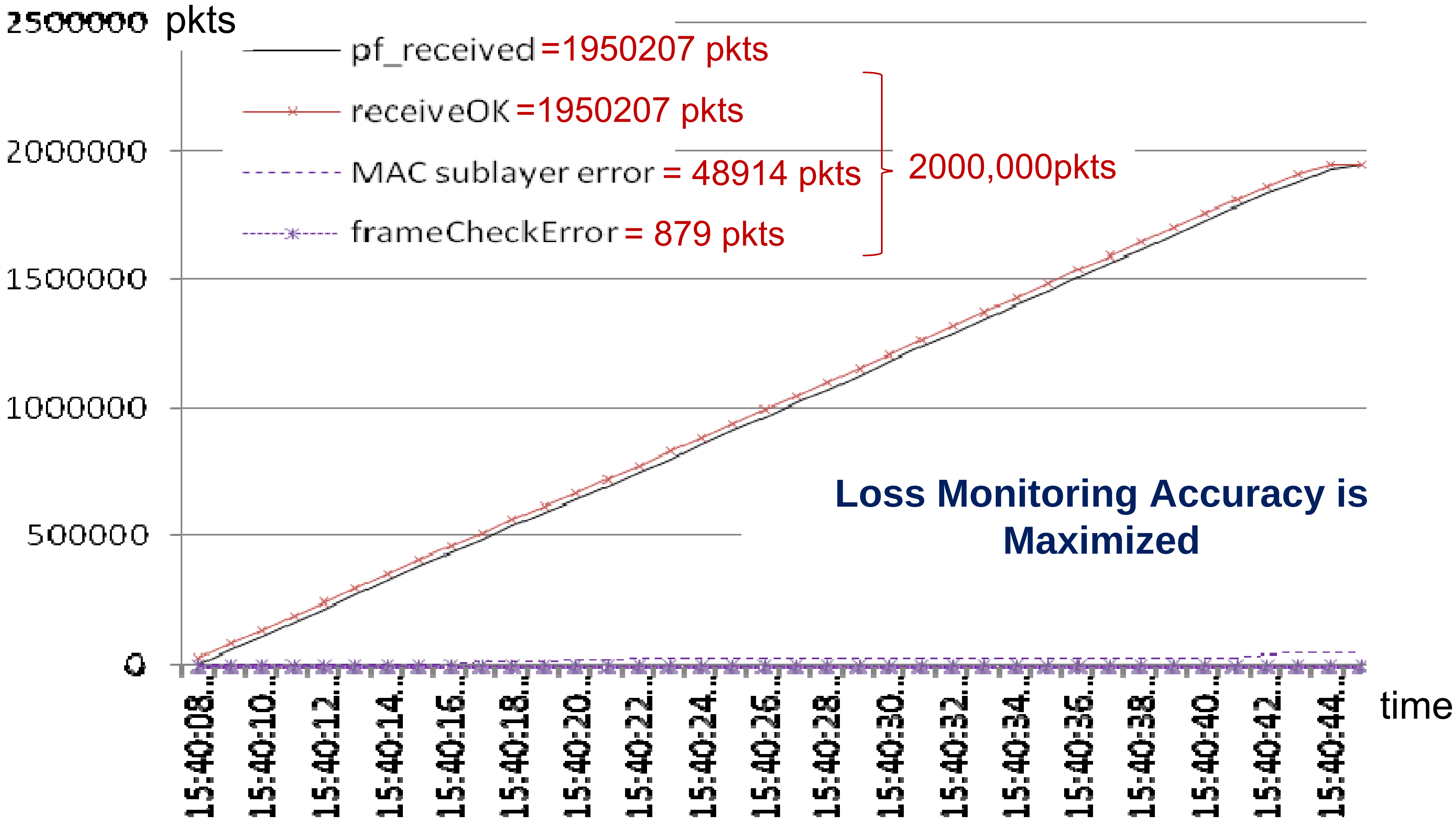
Reasonable overhead introduced by the our system wrapper (see gray-shaded data comparisons)

Performance Evaluation

Maximizing Packet Loss Monitoring Accuracy

Tests description:

- Metrics monitoring using PF_RING counters AND our system wrapper in parallel.
- 2 Million packets sent through a damaged cable.



Loss Monitoring Accuracy is Maximized

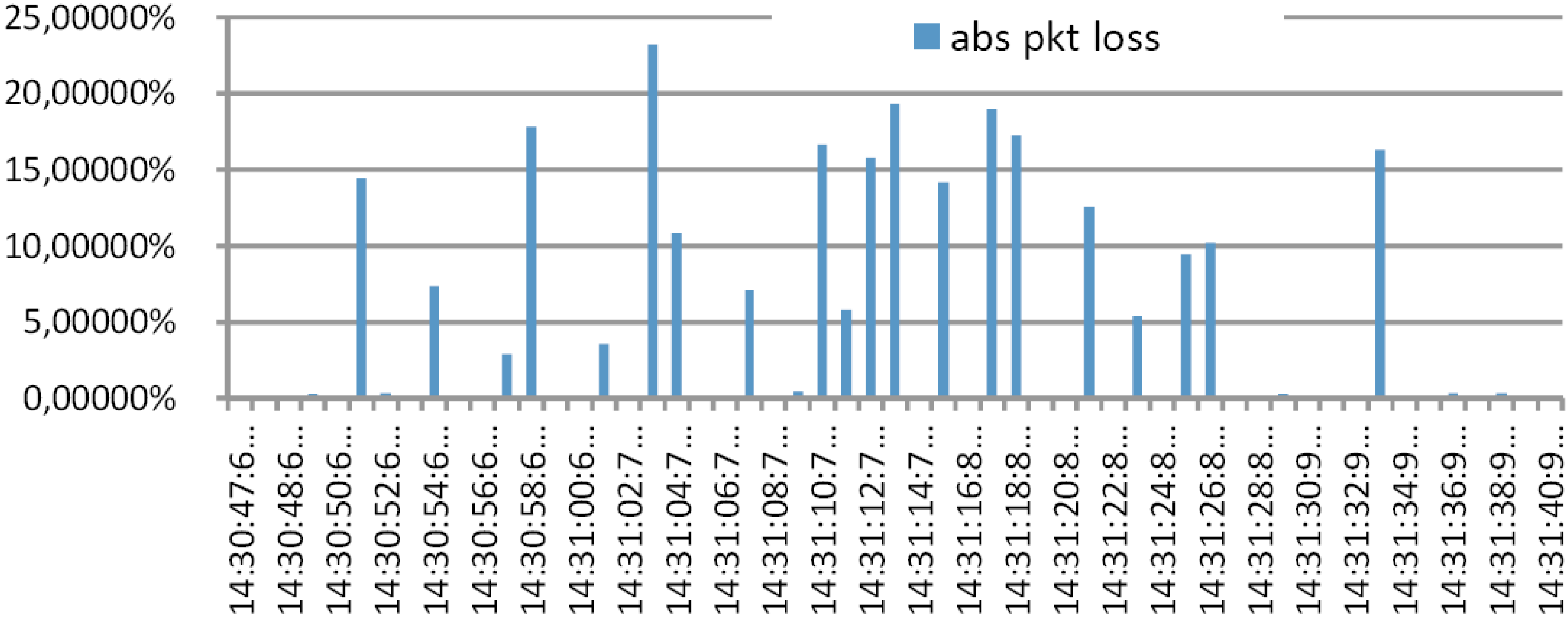
Performance Evaluation

Packet Loss Traceability through post-processing

Tests description:

- 2 Million packets of 1300Bytes sent through a damaged cable
- Tcpdump reported 1,878,123 pkts received with zero packet drops
- The post-processing step also counted the 1,878,123 pkts received with the following packet losses throughout the packet capture:

Post processing of the meta-traces allows validate the trace collections due to high or low packet losses



Discussion and Future Work

- The approach can be used as validation tool for the capturing set up

- Different levels of monitoring periodicity can be defined

- Fixed packet sizes and rates have been used to test the approach. Future work will be devoted to evaluate more realistic scenarios
- Other commodity NIC manufacturers and specific purpose hardware is also part of our future work
- 802.3 packet traces have been evaluated. The application of this method to wireless captures where MAC errors are likely to occur is part of our Future Work.

Summary conclusions

- We have presented an approach to maximize the accuracy of loss monitoring:
  - We presented an efficient method to retrieve IEEE Standard counters from Ethernet equipment
  - Metadata and meta-trace collection is an efficient instrument to validate the reliability of packet trace collections
  - Also, they provide the means to identify the exact location and the amount of packet losses exhibited by the capturing facility over the capturing process
  - Still, the accuracy of the loss monitoring process is limited to the actual counters implemented by the IEEE 802.3-compliant devices
  - The maximization of the packet loss monitoring process is affordable
  - (code for the loss monitoring facility and the post-processing available upon request)