User-level High-performance flexible network traffic processing library

Yanchao Jing
School of Shanghai
Jiao Tong University
Email:13541333146@163.com

Guangtao Xue
School of Shanghai
Jiao Tong University
Email: homer@thesimpsons.com

ShiyouQian
School of Shanghai
Jiao Tong University
Email:

***Abstract*—The abstract goes here.**

I. Introduction

Nowadays the ever-increasing demand for quality of service (Qos) and network security including policy-based routing,firewall and virtual private network(VPN) [1],In order to satisfy the requirement,An in-depth understanding of the Internet traffic profile is needed.Accordingly, the deep packet inspection(DPI) become a crucial hot technology[2].In a nutshell,a DPI system Firstly has to collect packets from the network interface cards(NIC),in order to create a data structure to represent the incoming packets as network flows and then to forward or store the received packets for further processing.So it is extremely important to capture all exchanged packets between end-systems on the monitored link in very high speed links (e.g. multiple Gbps).[2].many DPI application increasely high-level entities and constructs such as TCP flows Udp infomation HTTP headers and so on.Therefore a high-performance universal network traffic capture library is required to develop DPI high-level applications expediently.

Recently,For high-performance,enormous relative thoeritical methods and tools have been proposed such as zero-copy technology, multi-core capture,multi-thread method and FPGA tools and so on. FPGA tools is a optimization method of add-on cards[3] which is not in our focus scope.This paper mainly focus on software optimization; the zero-copy and multi-core technology can help avoid sparse sampling[4].A famous tool to finish these optimization is PF_RING[5].PF_RING kernel module runs as a software interrupt handler that stores incoming packets to a memeory-mapped buffer,shared with user-level stub. there are also many multi-thread proposed methods that can improve the entire performance. the core function of multi-thread is to make the best of multi-core performance.Although there are many kinds of theoretical technologies, there is hardly a flexible high-performance network traffic library. In this situation, developing a strong DPI application becomes a huge challenge.

To overcome this diffculty, in this paper we present a network traffic library(noff),a high-performance passive network traffic monitoring framework.noff provides the high-level functionality needed by monitoring application and provide a quite flexible framework to adjust some functions.Application developers can easily create magic high-level applications based on noff.

To acquire aggressive performance optimization, we use a flexible sur-level multi-thread structure towards a Multi queue NIC or just plain NIC.It can make best of the multi-core performance to achieve a high throughput.

To reduce the overhead of unneccessary memory copying.The noff regard the PF_RING as underlying kernel component. the subzero packet copy of pf_ring can contribute to reduce the total overhead and noff use google tcmalloc[] framework to reduce the malloc frequency.

Apparently, In order to synchronic the multi-thread system, mutex is a directly solution.But lock can bring implicative performance loss. noff make use of intel concurrency queue to make the entire system lockfree and get higher performance.

To accommodate heavy loads,noff introduces the notion of first input first output(FIFO) stream management algorithm.Under heavy load,traditional monitoring systems usually drop arriving packets in a random way,severely affecting any following stream reassembly process[].noff always drop the stream which is firstly built connection in hash table.this method can relatively reduce the impact from dropping packets.

Noff also provides a flexible and expressive Application programming interface that allows programmers to conifgure many kinds of stream aspect capture process procedure.Our design introduces two novel features:(I)noff use C++11 bind and function features to design callback interfaces between two modules.Programmers can flexiblely add new modules or change modules just in main config function rather than in module body.and(II)it offers more control for tolerating packet loss under high load throughput. And noff also support single queue NIC multi-thread scaling and multi queue based on RSS(receive-side scaling) transparent parallelization of stream processing simultaneously.

We have evaluated noff in a 10GbE environment using real compus traffic.and showed that it outperforms existing alternatives like Libnids and Snort's stream5 in variety of scopes.For example,our results demonstrate that noff can capture and deliver all streams with lower CPU utilization. The main contributions of this paper are:

1.We propose the noff,which is a universal network traffic library.It provides a flexible and aggressive module framework to offer programmers a easy-use platform to develop network

plug-in componentents.
2.We introduce some applied software techniques to optimize traditional products and approaches. through these techniques,noff can deliver transport-layer streams for two times higher traffic rates than previous approaches.
3.We introduce first in first out(FIFO),a algorithm that enables graceful adaptation to overload conditions by dropping packets of old streams,and favoring packets that belong to recent and shorter streams.

II. SOFTWARE FEATURE

The design of noff fouces attention on two key objectives: structure flexbility and runtime performance.In this part,we introduce the main aspects of noff across these two dimensions.

A. Flexble Module Framework and Interface

noff adopts a novel callback module interface which is based on C++11 bind and function feature.noff is composed of serveral modules. these modules can be divided into serveral parts by function or logical structure. the logical structure includes capturer,dispatcher,ip,tcp,udp,app and so on,which is correspond to Tcp/Ip network model to some extent.each layer will offer one or more callback interface to many upper layer modules.programmers can add their module based on lower layer module without change any existing code.At the same time,noff can use different modules at a same layer to adapt to different demand.

B. Pf_ring Patch

Pf_ring is a new type of network socket that dramatically improves the packet capture speed.It use User-space User-space ZC (new generation DNA, Direct NIC Access) drivers for extreme packet capture/transmission speed as the NIC NPU (Network Process Unit) is pushing/getting packets to/from userland without any kernel intervention. noff use Pf_ring as the capture driver and arrive aggressive runtime speed.

C. First Input First Output

noff introduces First Input First Output(FIFO) stream algorithm enable the system to invest its resources effectively during overload. This algorithm can resolve the problem that sudden traffic bursts or overload conditions may force the packet capturing subsystem to fill up its buffers and randomly drop packets in a haphazard manner.noff maintains a connection timer queue which sort the stream based on the first connection time in ascending order.when the stream buffer is full. noff would drop the stream in the head of timer queue.As long as the percentage of used memory is below a user-defines threshold,FIFO drops no packets.In this way,the newly stream will be accommodated with higher probability.Programmers can also force to maintain some priority streams by update the timestamp.

D. Flexible Stream reassembly

Libnids accomplished its Tcp Reassemble machine by simplify the linux kernel code. However,the callback interface of libnids is poor and it has no ability to cope with SYN Flood attack and connection abnormity because of none timer. Noff add a passive keep-alive timer to close the dead abnormal connection when a stream connection has no data interaction during a long time. Noff also abandon the tcp buffer to reduce memeory copy. Noff reserves nearly all of the function in libnids.

E. Memory Allocations Optimization

Noff uses google Tcmalloc to optimize the memeory copy performance. The Tcmalloc implementation takes approximately 50 nanoseconds for the same operation pair when ptmalloc2 takes approximately 300. The Tcmalloc assigns each thread a thread-local cache. Small allocations are satisfied from the thread-local cache. Objects are moved from central data structures into a thread-local cache as needed, and periodic garbage collections are used to migrate memory back from a thread-local cache into the central data structures.

F. Parallel Processing

1) Subsubsection Heading Here: Subsubsection text here.

III. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] Liu, Duo, et al. "High-performance packet classification algorithm for many-core and multithreaded network processor." International Conference on Compilers, Architecture and Synthesis for Embedded Systems ACM, 2006:334-344.

[2] Antonello, Rafael, et al. "Deep packet inspection tools and techniques in commodity platforms: Challenges and trends." Journal of Network & Computer Applications 35.6(2012):1863-1878.

[3] Qiao, Siyi, et al. "Network recorder and player: FPGA-based network traffic capture and replay." International Conference on Field-Programmable Technology IEEE, 2015.

[4] Ali, Sardar, et al. "On mitigating sampling-induced accuracy loss in traffic anomaly detection systems." ACM SIGCOMM Computer Communication Review 40.3(2010):4-16.

[5] PF_RING. http://www.ntop.org/products/packet-capture/pf_ring/