# Reliable Network Traffic Collection for Network Characterization and User Behavior

Ali Ismail Awad
Electrical Engineering Dept.,
Al Azhar University
Qena, Egypt
Email: aawad@ieee.org

Hanafy Mahmud Ali
Electrical Engineering Dept.,
Minia University
Minia, Egypt
Email: hanafy_mh@yahoo.com

Heshasm F. A. Hamed
Electrical Engineering Dept.,
Minia University
Minia, Egypt
Email: hfah66@yahoo.com

***Abstract*—This paper presents a reliable and complete traffic collection facility as a first and crucial step toward accurate traffic analysis for network characterization and user behavior. The key contribution is to produce an accurate, reliable and high fidelity traffic traces as the valuable source of information in the passive traffic analysis approach. In order to guarantee the traces reliability, we first detect the bottlenecks of the collection facility, and then propose different monitoring probes starting from the ethernet network interface and ending at the packet trace. The proposed facility can run without stop for long time instead of one-shot periods, therefore, it can be used to draw a complete picture of network traffic that fully characterize the network and user behavior. The laboratory experiments conclude that the system is highly reliable, stable and produces reliable traces attached with different statistics reports that come from the installed monitoring probes.**

## I. Introduction

Presently, Internet supports wide variety of applications via many protocol architecture instead of just data transfer. For example, data, voice signals, images and videos are supported by the same network infrastructure [1]. Due to the mixing nature of network traffic with targeted high speed connections, the understanding of the traffic behavior has become a difficult task. Traffic collection and analysis is considered as the right way for the network understanding and management [2].

Passively collected traffic traces include huge amount of information that is useful for the measuring of almost all network related activities. The analysis of packet traces provides information from user, network and service perspectives. It allows the identification and measurement of general trends of many different metrics useful for engineering, management and provisioning of the gigabit ethernet networks. The accuracy and reliability of the collected traffic traces have a direct impact on the outcome of different trace-based operations such as network characterization, traffic engineering [3], traffic modeling and user behavior estimation [4].

The accuracy and the reliability are two key issues of passive traffic collection. Collecting reliable packet traces without packet loss can be a difficult operation on gigabit ethernet networks under the usage of commodity based hardware and software. Conducting traffic analysis over incomplete and unreliable traffic traces leads to inaccurate results unless data losses are explicitly considered before the analysis process [5].

According to the resource constrains, the available collection facilities collect only one-shot of the network traffic that does not contain enough amount of information that reflexes the accurate characterization of the network. Additionally, these collection systems do not provide any reliability reports about the collected traces, and hence, the analysis results of these traces may be inaccurate and unreliable. A reliable packet capturing facility must be equipped with a mechanism to accurately report the time and amount of packet loss during the trace collection operation [6].

The usage of on-the-shelf hardware and software for packet capturing on a high-speed (1 Gbps or higher) is sensible to packet losses. Most of the carried out researches with the commodity equipments is directed toward enhancing the performance of packet capturing with respect to software [7], and hardware [8] in order to cope the network line speed [9]. Data Acquisition and Generation (DAG) [10] is a dedicated hardware solution for reliable packet capturing on high-speed networks with high cost compared to the commodity solutions.

This paper focuses on the reliability of the collection facility, and presents a reliable and complete traffic collection facility using commodity hardware and softwares. The efficient usage of the produced facility provides a very useful information for different network users [4]. Network users be categorized into Internet Service Providers (ISPs), devices and hardware manufacturers, network administrations and network researchers. ISPs use traffic analysis results for billing their customers, identifying the dominant applications, and hence they can build an accurate Service Level Agreement (SLA)[11]. Moreover, ISPs can use the traffic analysis for network management, provisioning and troubleshooting recovery. Hardware providers use traffic analysis results for measuring devices behavior under different conditions, and hence they can make decisions to enhance or redesign the current network devices. Traffic analysis will be useful for network administrators to detect the up normal behavior of the network traffic. Researchers use network traffic analysis to understand and developing different traffic models.

The reminder part of this paper is organized as follows. Section II demonstrates the structure of the generic collection facility with bottleneck points, and emphasising the proposed network interface monitoring approach. Section III explains the implementation of the network interface monitoring approach. Section IV shows the exhaustive evaluation of the reliable facility in terms of resources overhead and accuracy. Conclusions and future work are reported in section V.

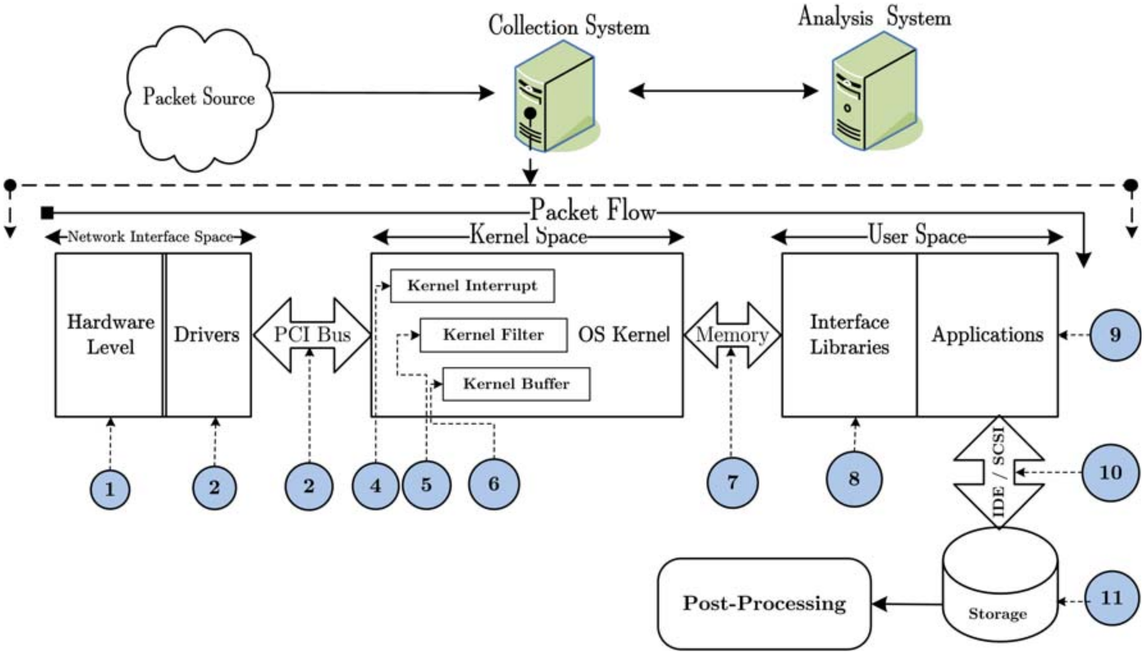

Fig. 1. A Complete traffic collection facility with bottleneck points marked in circles. The bottlenecks diffuse in almost all system spaces.

## II. Reliable Network Traffic Collection

This research focuses on the passive traffic measurement methodology to address the traffic collection and analysis problem in the residential networks. However, it is not easy to do continuous passive measurement, but on the other hand, passive measurement provides a complete picture about network traffic. Moreover, different traffic analysis phases can be conducted on the traffic traces from different perspectives. The performance metric of any passive collection is the lossless packet capturing at link speed. Dropped packet will produce problem in the next processes (anonymization, analysis, etc.). Each component in the collection facility has its own characteristics and limitations. Therefore, the bottlenecks are distributed over all components. Fig. 1 shows the bottlenecks through the packet journey through all collection facility components.

A closed look at Fig. 1 shows that the traffic collection facility suffers from many bottleneck points starting from network interface, passing though kernel space, and ends with the packet capturing applications in the user space. Every system bottleneck point can lead to packet drops without any feedback information to the system operator. Unreported packet drops deteriorates the reliability of the traffic collection facility especially at the 1 Gbps link speed.

The available solutions of the collection bottlenecks are designed to overcome one point to enhance the collection performance in terms of packet loss and scalability to link speed. Those solutions can be divided into software-based such as Driverdump [12] and Interrupt Coalescence (IC) technique [13], kernel patching [14] and hardware-based solutions such as Network Processor (NP) [8], [15], [16], and special purpose DAG card [10]. Some solutions try to build special purpose facilities, but those solutions are cost inefficient. The problem with the previous solutions is that they have been designed for special purpose or enhancing a particular point. Additionally, the implementations of those solutions may become difficult due to some coding problems or the high cost. We have built a new solution for bottlenecks and insure the system reliability by installing monitoring probe for each system bottleneck using commodity based hardware and software. The most important probe is the network interface monitoring approach.

### A. Network Interface Monitoring Approach

Network interface card is the first contact point inside the collection machine that can hold all packets including the correct and the erroneous ones at 1 Gbps link speed. Network interface monitoring approach uses network interface capability to monitor all coming in and out packets to the collection machine before its pumping up to the upper levels. We could correlate the produced report with the trace file in the post processing to detect the packet loss, and judging the packet trace reliability for the collection session.

The idea behind monitoring the ethernet network interface is considered as two folded process: (1) Open a socket for direct communication with the network interface drivers, and (2) Information exchange between network interface drivers and the monitoring tool agent in the user interface. In order to retrieve statistics directly from hardware, the proposed monitoring approach takes advantage of the support provided by the Ethtool Linux utility [17]. Ethtool is a GNU/Linux tool that allows obtaining information and diagnostics about ethernet card settings related to media, link status, and more. Precisely, the `ethtool_stats` data structure provided by its API enables dumping the network interface specific statistics to the user space, and store them in a statistics file.

## III. Implementation Phase

The proposed monitoring approach has been fully implemented with the C programming language and under Linux environment. Beside different additional functions, the implemented core function is `do_gstats()` which is responsible for consulting the network interface hardware via its drivers, open a User Datagram Protocol (UDP) socket and retrieving back the available statistics.

The simplified interconnections diagram of the implemented functions is shown in Fig. 2. The `main()` function has a direct connection to the `do_metatrace()` which is responsible for creating the output file name, print file headers and arrange the spaces between columns. The time stamp is calculated using `delta_time()` before each hardware check. The `delta_time()` has a time stamp sensitivity up to 1 microsecond, therefore, the proposed approach is able to read the network interface hardware statistics values every 1 microsecond. The function `do_print()` is used for printing the output results to a statistics file or directly to the screen based on the way of its call and the passed parameters. The function `main()` can also directly call `do_gstats()` and print the results directly to the screen instead of writing it to an output file.

### A. The Function do_gstats()

The function `do_gstats()`is the most important one inside the implementation structure, and it is declared as `do_gstats(char *ifname, int s_order)`. While `ifname` is a pointer to the network interface name, and `s_order` is the order of the statistic indicator inside the data array. The `do_gstats()` function returns an `unsigned long long` statistic values depend on the input parameter `s_order`. The sequence of `do_gstats()` instructions starts with setting all parameters and access all data structures, then open a UDP data socket through network interface drivers to access hardware statistics. Through the opened socket, the hardware statistics are dumped to an array, then the `do_gstats()` returns the selected element to be recorded in the statistics file. Fig. 2 shows the calling methods of `do_gstats()` and its relation with the other functions. As a general remark, in order to implement the method outlined in this section, the implementation code should include `<linux/netdevice.h>`, `<linux/etherdevice.h>`, and `<linux/ethtool.h>` Linux headers files.

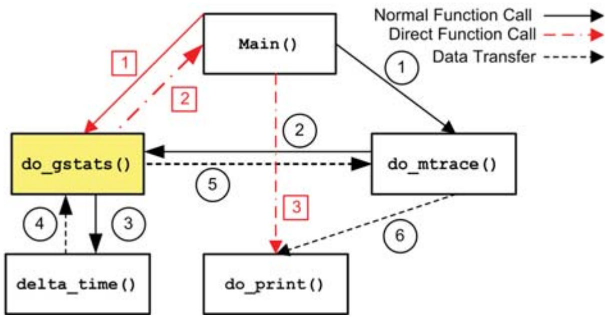

Fig. 2. Functions interconnection diagram of the monitoring approach

## IV. Experimental Evaluation

The presented results in this section have been obtained from experiments that have been conducted in a controlled environment constructed from one PC and one Laptop. The PC works as a traffic collector and equipped with Intel Pentium® 4 Processor 3 GHz, 1 GB of RAM, Intel gigabit ethernet card, and 160 GB hard disk. The Laptop works as traffic generator and equipped with Intel Core 2 Due™ 2.5 GHz Processor, 4 GB of RAM, and Intel Gigabit Ethernet card. The two computers are directly connected through a gigabit ethernet cards via special type UDP cable. Both machines have been equipped with an implementation of the network interface monitoring approach.

The collector machine has been installed with Ubuntu Linux kernel $2.16.18 - 1.2200$, PF_RING patched kernel [14], Tcpdump version 3.9.4 with Libpcap version 0.9.4 [18]. It is worth noticing that Libpcap has been recompiled with the PF_RING toolkit modifications, also the Tcpdump has been recompiled against the PF_RING modified by Libpcap. The generator machine has been installed with the same Ubuntu Linux kernel. The traffic is generated with the open source PackETH as a packet generator toolkit [19].

### A. Monitoring Approach Overhead Test

We first considered the CPU overhead introduced by the periodical (1 second) monitoring granularity. We have run two independent tests: (I) Tcpdump packet capturing with monitoring approach enabled, and (II) Tcpdump capturing with proposed loss monitoring enabled. We have sent a fixed amount of generated packets, (2 millions packets), into the collector machine. Fig.3 shows the CPU utilization of both experimental scenarios. From that figure, running the monitoring technique in parallel with Tcpdump does not introduce high extra CPU overhead, and hence, the proposed monitoring approach does not provide a resource limitation at the network interface saturation point.

### B. Facility Overall Accuracy Test

This test is carried out to check the facility performance with enhanced Linux kernel using PF_RING explained in [14]. Fig. 4 shows the generated packet rates for each packet

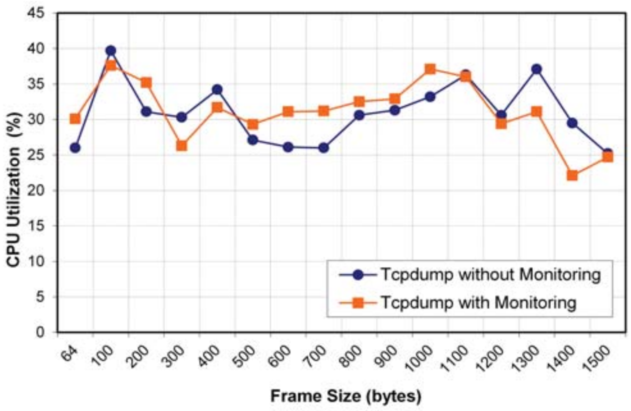

Fig. 3. CPU utilizations for one collection session with monitoring approach.

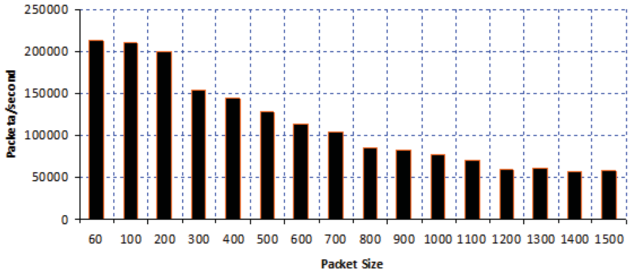

Fig. 4. Packet generation rates measured in generator side.

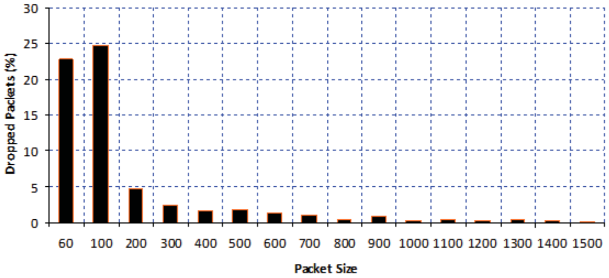

Fig. 6. The accuracy of the collection facility in terms of packet loss.

sizes. The plotted data have been taken from the statistics file produce by the monitoring approach in the traffic generator side. Additionally, from the collector point of view, Fig. 5 shows the percentage of the collected packets for each packet size. The figure proves the high packet losses at short packet sizes. The overall accuracy can be predicted by finding the difference between the generated and the collected packets. Of course, the packet trace reliability is determined according to the amount of dropped packets compared to the generated ones. The dropped packets are directly reported by the monitoring approach (values in the statistic files). While the kernel drops are measured as the difference between captured packets and packets received by the network interface (deduced from statistics information). From Fig. 6, the proposed monitoring approach is always reporting the total generated packets (received + erroneous) with 100% accuracy, and hence, the reliability of the packet trace can be measured by correlating the reported packets by the proposed approach and the actual collected packets in the trace file.

### C. Discussion

Although the related work shows many special purpose hardware and software solutions for packet drop problem, the presented results in this section prove the superiority of using commodity based hardware and software in the proposed solution with invented monitoring probes at every bottleneck in the collection facility. The statistic files produced by each monitoring probe can be correlated with the actual recorded packets in the trace file, and hence, we get knowledge about where and when the packet was dropped. The actual packet trace is then sanitated further to remove the gaps of the dropped packets which finally produces a reliable packet trace.

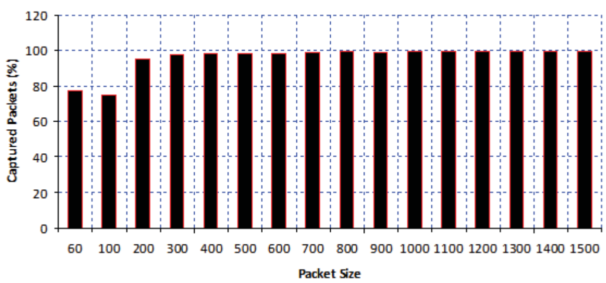

Fig. 5. Packet collection percentage (%) measured in collector side.

Once we got a reliable and accurate passively collected packet trace with attached monitoring reports, the traffic analysis for network characterization and user behavior will be the next phase of this research.

## V. Conclusion and Future Work

This paper was directed toward enhancing the reliability of traffic collection facility. It has presented a new network interface monitoring approach in order to increase the reliability of the collected packet trace. The evaluation results have proved that the proposed mechanism is non-intrusive for the traffic trace collection with respect to CPU consumption, packet generation and packet collection. The experimental works conclude that purposed monitoring approach is a feasible and practical tool for reliable packet trace collection. As a future work, the proposed approach can be extended for different ethernet cards and implemented for wire and wireless network interfaces on different Linux platforms.

## Acknowledgement

The initial design and implementation of this work has been conducted in the laboratory of NetCom, Universitat Pompeu Fabra, Barcelona, Spain. I would like to thank Professor Dolors Sala for her advices during building the initial phase of this research.

## References

[1] J. Goldman and P. Rawles, *Local area networks: a business-oriented approach*, 2nd ed. John Wiley & Sons, 2000.

[2] J. Rubio-Loyola, D. Sala, and A. I. Ali, “Maximizing packet loss monitoring accuracy for reliable trace collections,” in *Proceedings of the* $16^{th}$ *IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2008)*. Chij-Napoca, Romania: IEEE, 2008, pp. 61–66.

[3] B. Eriksson, P. Barford, and R. Nowak, “Network discovery from passive measurements,” *Computer Communication Review*, vol. 38, no. 4, pp. 291–302, 2008.

[4] J. L. Jerkins and J. L. Wang, “A close look at traffic measurements from packet networks,” in *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*, vol. 4, 1998, pp. 2405–2411.

[5] A. I. Awad, H. M. Ali, and H. F. A. Hamed, “Toward highly reliable network traffic traces,” in *Proceedings of the First International Conference on Communications, Signal Processing, and their Applications, ICCSPA13*. Sharjah, United Arab Emirates: IEEE, February 2013, p. To Appear.

[6] J. Rubio-Loyola, D. Sala, and A. I. Ali, “Accurate real-time monitoring of bottlenecks and performance of packet trace collection,” in *Proceedings of the* $33^{rd}$ *IEEE Conference onLocal Computer Networks (LCN 2008)*. Montreal, Que, Canada: IEEE, 2008, pp. 884–891.

[7] G. Iannaccone, C. Diot, I. Graham, and N. McKeown, “Monitoring very high speed links,” in *Proceedings of the* $1^{st}$ *ACM SIGCOMM Workshop on Internet Measurement*. San Francisco, California, USA: ACM, 2001, pp. 267–271.

[8] R. Ramaswamy, N. Weng, and T. Wolf, “A network processor based passive measurement node,” in *Proceedings of the* $6^{th}$ *international conference on Passive and Active Network Measurement (PAM’05)*. Boston, MA: Springer-Verlag, 2005, pp. 337–340.

[9] E. Weigle and W. chun Feng, “TICKETing high-speed traffic with commodity hardware and software,” in *Proceedings of the Third Annual Passive and Active Measurement Workshop (PAM2002)*, 2002, pp. 156–166.

[10] “Data Acquisition and Generation (DAG).” [Online]. Available: http://www.endace.com/

[11] W. Stallings, *Data & Computer Communications*, six ed. Prentice Hall, 1999.

[12] E. Anderson and M. Arlitt, “Full packet capture and offline analysis on 1 and 10 gb/s networks,” Technical Report, HPL-2006-156 20061106, HP Labs, Tech. Rep., 2006.

[13] R. Prasad, M. Jain, and C. Dovrolis, “Effects of interrupt coalescence on network measurements,” in *The* $5^{th}$ *anuual Passive & Active Measurement Workshop, (PAM 2004)*, Antibes, France, April 2004.

[14] L. Deri, “Improving passive packet capture: Beyond device polling,” in *Proceedings of SANE 2004*, 2004.

[15] K. Mackenzie, W. Shi, A. Mcdonald, and I. Ganev, “An intel IXP1200-based network interface,” in *Proceedings of the Workshop on Novel Uses of System Area Networks at HPCA (SAN-2 2003)*, 2003.

[16] T. Nguyen, M. Cristea, W. de Bruijn, and H. Bos, “Scalable network monitors for high-speed links: a bottom-up approach,” in *Proceedings IEEE Workshop on IP Operations and Management, 2004.*, Beijing, China, October 2004, pp. 16–22.

[17] “Free software directory. the ethtool resource: a net driver diagnostic and tuning tool.” [Online]. Available: http://directory.fsf.org/project/ethtool/

[18] “Berkley Packet Filter, Lawrence Berkeley National Laboratory Network Research. TCPDump: the Protocol Packet Capture and Dumper Program.” [Online]. Available: http://www.tcpdump.orgmp.org

[19] M. Jemec, “PackETH, Open Source Ethernet Packet Generator.” [Online]. Available: http://packeth.sourceforge.net/