www.ijiarec.com

ISSN:2348-2079

Volume-8 Issue-3

International Journal of Intellectual Advancements and Research in Engineering Computations

Traffic classification based on zero-length packets

Dr.R.Punidha[1], V.Ragavi[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Bharathiyar Institute of Engineering for Women, Deviyakurichi, Attur- 636112

[2]PG Scholar Master of Engineering, Department of Computer Science and Engineering, Bharathiyar Institute of Engineering for Women, Deviyakurichi, Attur- 636112

ABSTRACT

Network traffic classification is fundamental to network management and its performance. However, traditional approaches for traffic classification, which were designed to work on a dedicated hardware at very high line rates, may not function well in a virtual software-based environment. In this paper, we devise a novel fingerprinting technique that can be utilized as a software-based solution which enables machine-learning based classification of ongoing flows. The suggested scheme is very simple to implement and requires minimal resources, yet attains very high accuracy. Specifically, for TCP flows, we suggest a fingerprint that is based on zero-length packets, hence enables a highly efficient sampling strategy which can be adopted with a single CAM rule. The suggested fingerprinting scheme is robust to network conditions such as congestion, fragmentation, delay, retransmissions, duplication and losses and to varying processing capabilities. Hence, its performance is essentially independent of placement and migration issues, and thus yields an attractive solution for virtualized software-based environments. We suggest an analogous fingerprinting scheme for UDP traffic, which benefits from the same advantages as the TCP one and attains very high accuracy as well. Results show that our scheme correctly classified about 97% of the flows on the dataset tested, even on encrypted data.

Keywords: Very High Accuracy, Machine-Learning, Network Traffic Classification.

INTRODUCTION

Measuring and analyzing network traffic dynamics between end hosts has provided the foundation for the development of many different network protocols and systems. Of particular importance understands packet loss behavior since loss can have a significant impact on the performance of both TCP- and UDP-based applications. Despite efforts of network engineers and operators to limit loss, it will probably never be eliminated due to the intrinsic dynamics and scaling properties of traffic in packet switched network. Network operators have the ability to passively monitor nodes within their network for packet loss on routers using SNMP. End-to-end active measurements using probes provide an equally valuable perspective since they indicate the conditions that application traffic is experiencing on those paths. There are trade-offs in packet loss measurements between probe rate, measurement accuracy, impact on the path and timeliness of results. The objective is to accurately measure loss characteristics on end-to-end paths with probes [1-5].

Measuring and analyzing network traffic dynamics between end hosts has provided the foundation for the development of many different network protocols and systems. Of particular importance understands packet loss behavior since loss can have a significant impact on the performance of both TCP- and UDP-based applications. Despite efforts of network engineers

Author for correspondence:
Department of Computer Science and Engineering, Bharathiyar Institute of Engineering for Women, Deviyakurichi, Attur- 636112

and operators to limit loss, it will probably never be eliminated due to the intrinsic dynamics and scaling properties of traffic in packet switched network. Network operators have the ability to passively monitor nodes within their network for packet loss on routers using SNMP. End-to-end active measurements using probes provide an equally valuable perspective since they indicate the conditions that application traffic is experiencing on those paths [6-10].

The most commonly used tools for probing end-to-end paths to measure packet loss resemble the ubiquitous PING utility. PING-like tools send probe packets (e.g., ICMP echo packets) to a target host at fixed intervals. Loss is inferred by the sender if the response packets expected from the target host are not received within a specified time period. Generally speaking, an active measurement approach is problematic because of the discrete sampling nature of the probe process. Thus, the accuracy of the resulting measurements depends both on the characteristics and interpretation of the sampling process as well as the characteristics of the underlying loss process.

RELATED WORK

To support a vast range of network applications, SDN has been designed to apply flow-based rules, which are more complex than destination-based rules in traditional IP routers. For instance, access-control requires matching on source, destination IP addresses, port numbers and protocol whereas a load balancer may require matching's only on source and destination IP prefixes. These complicated matching's can be well supported using TCAM since all rules can be read in parallel to identify the matching entries for each packet. However, as TCAM is expensive and extremely power-hungry, the on-chip TCAM size is typically limited. Many existing studies in the literature have tried to address this limited rule space problem. For instance, the authors in and try to compact the rules by reducing the number of bits describing a flow within the switch by inserting a small tag in the packet header. This solution is complementary to ours, however, it requires a change in: (i) packet headers and (ii) in the way the SDN tables are populated. Also, adding an identifier to each incoming packet is hard to be done in the ASICs since this is not a standard operation, causing the packets to be processed by the CPU (a.k.a. the slow-path), strongly penalizing the performance of a forwarding device and the traffic rate. Another approach is to compress policies on a single switch. For example, the authors in have proposed algorithms to reduce the number of rules required to realize policies on a single switch.

LITERATURE SURVEY

M. Rifai, N. Huin, C. Caillouet, F. Giroire, D. Lopez-Pacheco, J. Moulierac, and G. Urvoy-Keller

Software Defined Networking (SDN) is gaining momentum with the support of major manufacturers. While it brings flexibility in the management of flows within the data center fabric, this flexibility comes at the cost of smaller routing table capacities. In this paper, we investigate compression techniques to reduce the forwarding information base (FIB) of SDN switches. We validate our algorithm, called MINNIE, on a real testbed able to emulate 20 switches fat tree architecture. We demonstrate that even with a small number of clients, the limit in terms of number of rules is reached if no compression is performed, increasing the delay of all new incoming flows. MINNIE, on the other hand, reduces drastically the number of rules that need to be stored with a limited impact on the packet loss rate. We also evaluate the actual switching and reconfiguration times and the delay introduced by the communications with the controller.

R. Alshammari and A. N. Zincir-Heywood

The objective of this work is to assess the robustness of machine learning based traffic classification for classifying encrypted traffic where SSH and Skype are taken as good representatives of encrypted traffic. Here what we mean by robustness is that the classifiers are trained on data from one network but tested on data from an entirely different network. To this end, five learning algorithms – AdaBoost, Support Vector Machine, Na¨ıve Bayesian, RIPPER and C4.5 – are evaluated using flow based features,

where IP addresses, source/destination ports and payload information are not employed. Results indicate the C4.5 based approach performs much better than other algorithms on the identification of both SSH and Skype traffic on totally different networks.

L. Grimaudo, M. Mellia, and E. Baralis

Traffic classification is still today a challenging problem given the ever evolving nature of the internet in which new protocols and applications arise at a constant pace. In the past, so called behavioral approaches have been successfully proposed as valid alternatives to traditional dpi based tools to properly classify traffic into few and coarse classes. In this paper we push forward the adoption of behavioral classifiers by engineering a hierarchical classifier that allows proper classification of traffic into more than twenty fine grained classes. Thorough engineering has been followed which considers both proper feature selection and testing seven different classification algorithms. Results obtained over actual and large data sets show that the proposed hierarchical classifier outperforms off-the-shelf nonhierarchical classification algorithms by exhibiting average accuracy higher than 90%, with precision and recall that are higher than 95% for most popular classes of traffic.

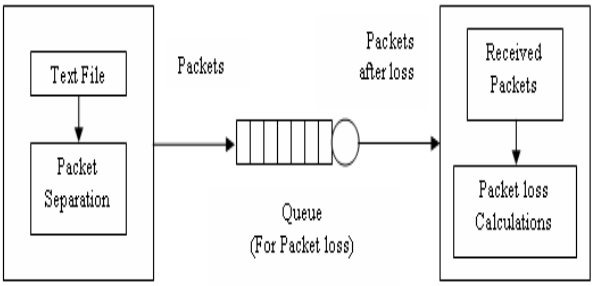
PROPOSED SYSTEM

We have designed, developed, and implemented a compromised router detection protocol that dynamically infers, based on measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Once the ambiguity from congestion is removed, subsequent packet losses can be attributed to malicious actions. We have tested our protocol in Emulab and have studied its effectiveness in differentiating attacks from legitimate network behavior.

Advantages

- We have designed, developed, and implemented.
- A compromised router detection protocol.
- We have tested our protocol in Emulab and have studied its effectiveness.
- In differentiating attacks from legitimate network behavior.

System Model



Modules

- User Interface Design
- Packet Separation
- Implementation of the Queue
- Packet Receiver
- Packet Loss Calculation

User interface design

In this module we design the user interface for Sender, Queue, Receiver and Result displaying window. These windows are designed in order to display all the processes in this project.

Packet Separation

In this module the data which we are selecting to send is divided into packets and then those sent to the Queue.

Designing the Queue

The Queue is designed in order to create the packet loss. The queue receives the packets from the Sender, creates the packet loss and then sends the remaining packets to the Receiver.

Packet receiver

The Packet Receiver is used to receive the packets from the Queue after the packet loss. Then the receiver displays the received packets from the Queue.

Packet loss calculation

The calculations to find the packet loss are done in this module. Thus we are developing the tool to find the packet loss.

METHODOLOGY

In this paper, we consider the problem of dynamically routing traffic demands inside a data center network using SDN technologies. In Section III, we provide the routing algorithm satisfying the link capacity constraints and the compression algorithm satisfying the routing table size constraints of the different forwarding devices. We validate our algorithms with a tested composed of SDN hardware as explained in Section IV. We study different metrics, namely the delay introduced by the communications with the controller, the potential increase of fault rate due to the handling of the dynamic routing and the load of the controller with and without compression. Our results (Section V and VI) show that we are able to minimize the number of entries in the switches, while successfully handling client's dynamics and keeping the network stability. Finally, we close this document in Section VII by providing concluding remarks and future works.

CONCLUSION

To the best of our knowledge, this paper is the first serious attempt to distinguish between a router dropping packets maliciously and a router dropping packets due to congestion. Previous work has approached this issue using a static user-defined threshold, which is fundamentally limiting.

Using the same framework as our earlier work (which is based on a static user-defined threshold), we developed a compromised router detection protocol that dynamically infers, based on measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Subsequent packet losses can be attributed to malicious actions. Because of non-determinism introduced by imperfectly synchronized clocks and scheduling delays, protocol uses user-defined significance levels, but these levels are independent of the properties of the traffic. Hence, protocol does not suffer from the limitations of static thresholds.

REFERENCES

[1]. B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar "The design and implementation of open vswitch." in NSDI, 2015, 117–130.

[2]. M. Rifai, N. Huin, C. Caillouet, F. Giroire, D. Lopez-Pacheco, J. Moulierac, and G. Urvoy-Keller, "Too many sdn rules? compress them with minnie," in Global Communications Conference (GLOBECOM), IEEE, 2015, 1–7.

[3]. C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a better netflow," in ACM SIGCOMM Computer Communication Review, 34(4). ACM, 2004, 245–256.

[4]. N. Hohn and D. Veitch, "Inverting sampled traffic," in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. ACM, 2003, 222–233.

[5]. N. Duffield, C. Lund, and M. Thorup, "Learn more, sample less: control of volume and variance in network measurement," Information Theory, IEEE Transactions on, 51(5), 2005, 1756–1775.

[6]. S. Fernandes, C. Kamienski, J. Kelner, D. Mariz, and D. Sadok, "A stratified traffic sampling methodology for seeing the big picture," Computer Networks, 52(14), 2008, 2677–2689.

[7]. S. Zander, T. Nguyen, and G. Armitage, "Sub-flow packet sampling for scalable ml classification of interactive traffic," in Local Computer Networks (LCN), IEEE 37th Conference on. IEEE, 2012.

[8]. N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "Rule-caching algorithms for software-defined networks," Technical report, 2014.

[9]. A. Vishnoi, R. Poddar, V. Mann, and S. Bhattacharya, "Effective switch memory management in openflow networks," in Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems. ACM, 2014, 177–188.

[10]. M. Kuzniar, P. Pere ´ sıni, and D. Kostic, "What you need to know about ´ sdn flow tables," in International Conference on Passive and Active Network Measurement. Springer, 2015, 347–359.