Revealing Hidden Hierarchical Heavy Hitters in network traffic

Salvator Galea
Gianni Antichi
Andrew W. Moore
University of Cambridge
Firstname.Lastname@cam.ac.uk

Giuseppe Bianchi
University of Rome Tor Vergata
giuseppe.bianchi@uniroma2.it

Roberto Bifulco
NEC Laboratories Europe
roberto.bifulco@neclab.eu

ABSTRACT

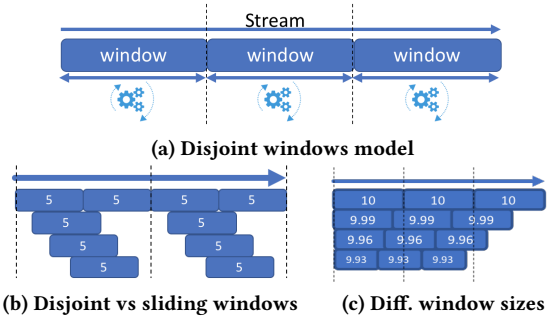
The idea to enable advanced in-network monitoring functionality has been lately fostered by the advent of massive data-plane programmability. A specific example includes the detection of traffic aggregates with programmable switches, i.e., heavy hitters. So far, proposed solutions implement the mining process by partitioning the network stream in disjoint windows. This practice allows efficient implementations but comes at a well-known cost: the results are tightly coupled with the traffic and window's characteristics.

This poster quantifies the limitations of disjoint time windows approaches by showing that they hardly cope with traffic dynamics. We report the results of our analysis and unveil that up to 34% of the total number of the hierarchical heavy hitters might not be detected with those approaches. This is a call for a new set of windowless-based algorithms to be implemented with the match-action paradigm.

1 INTRODUCTION

Data-plane programmability opened up new opportunities to improve current network management tasks, i.e., accounting, traffic engineering and anomaly or Distributed Denial-of-Service (DDoS) detection. In the past, it has been recognized that these operations can be effectively performed by detecting in real time the presence of high volume traffic clusters. In this scenario, the research community has been lately working on finding new solutions to enable in-network detection of heavy flows, leveraging the massive switch programmability offered by P4-based systems [4, 5].

Until now, characterizing the presence of high volume traffic aggregates has been tackled as a problem of detecting Heavy Hitter (HH) or Hierarchical Heavy Hitter (HHH). The former seeks to find an IP prefix p which contributes with a traffic volume larger than a given threshold T during a fixed time interval t. The latter is a special case of the former. Specifically, it looks for a prefix p, which exceeds a threshold T after excluding the contribution of all its HHH descendants. To simplify the online traffic analysis, most of the proposed solutions suggest to divide the network stream into fixed-time disjoint intervals (Figure 1a) and perform the required identification process in each of them separately, without considering the traffic trends from previous intervals [4, 5].



(a) Disjoint windows model

(b) Disjoint vs sliding windows

(c) Diff. window sizes

Figure 1: Window model and our analysis

This practice helps the efficiency of detection process: at the end of each time window, it is trivial to identify the flows that consume more than a fraction τ of the link capacity, i.e., heavy hitter. In addition, by resetting the data structure at the end of each time window, there is no risk of counter overflowing, which otherwise could lead to misleading results. Although past research has recognized the importance of adopting a different approach [1], it is not clear yet the trade-offs carried by disjoint window-based algorithms.

This poster is a first step towards a better understanding the limitation of common approaches to HH and HHH detection. By analyzing real traffic traces taken from a Tier-1 ISP, we quantify the limitation of disjoint window-based approaches. We eventually reveal the presence of *hidden* HHHs and we use such a study to call for a new set of match-action friendly algorithms that overcome nowadays limitations.

2 THE CASE FOR BETTER ANALYSIS

We aim to quantify the relation between the window's characteristic and its (H)HH detection capabilities. Specifically, we analysed real network traffic [3] (1 hour long traces taken from four different days) and we compared (i) the results produced by two window-based approaches, as well as (ii) the results of different window sizes (e.g. how much small variation of its dimension can impact the final result).

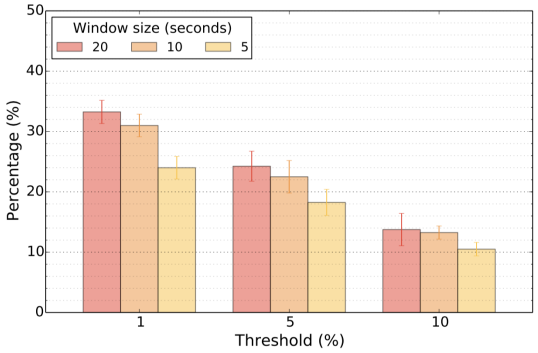


Figure 2: Percentage of hidden HHH for three different window sizes and thresholds.

Unveiling Hidden HHHs. In this experiment we quantify the difference between fixed-time disjoint windows and a sliding window approach. Specifically, we compared the outputs of 5, 10 and 20 seconds time windows against one that uses a sliding window of the same length (Fig. 1b). We consider HHH, the flows which exceed 1%, 5%, 10% of the total bytes measured in a specific time-window. The results show that approaches based on disjoint fixed time windows can hide many information. As depicted in Fig.2, we found that up to 34% of the total number of the HHH might not be detected by disjoint window approaches. This is for sure not acceptable as HHHs can be used for accounting, traffic engineering or DDoS attack detection. Moreover in all cases of window sizes, the hidden information accounts from 24% to 34% and 18% to 24% of the total number of HHHs, for 1% and 5% threshold respectively.

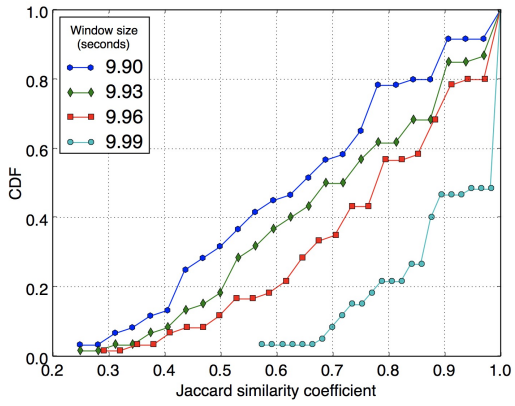


Figure 3: Similarities of reported HHHs to the baseline window

Micro variations in window sizes lead to different results. Using as a baseline a fixed time window of 10 seconds, we compare the detected HHHs against the one identified in other time windows which are 10-100 milliseconds shorter from the baseline window (Fig. 1c). All the windows have the same starting point and the analysis is based only on overlapping windows. The trace duration is 20 minutes and the HHHs account for the flows which exceed 5% of the traffic in each window. The results produced by the baseline window have been compared against the one obtained with different windows sizes using the Jaccard similarity coefficient. This metric estimates the diversity or the similarity of sets. Fig. 3 shows that window sizes of 100 and 40 milliseconds smaller than the baseline window differs by 25% and 11% respectively, for at least 70% of the cases.

3 TOWARDS TIME DECAYING ANALYSIS OF TRAFFIC

The previous experiments expose the fact that window's characteristics are interdependent with the aggregation process and its results. As the existence of hidden HHH has been revealed, we need to consider new directions to streaming algorithms which are based on continuous-time operation and can overcome the accuracy limitations of the original disjoint window approaches. More specifically, as a first step towards this evaluation, we consider to implement a Time-decaying Bloom Filter and its extension [2] as a proof of concept. Although we are open to other solutions, we have chosen that particular streaming algorithms for its simplicity. A goal for future work is to implement them on programmable data-plane devices. Eventually this will be an attempt to compare it with existing solutions in terms of performance, resource utilization and result's accuracy.

4 CONCLUSION

Our offline analysis reveals that *"hidden"* heavy flows account up to 34% of the total number of identified heavy flows in fixed time windows and for even small variations in the window size, their aggregates can significantly diverge. Consequently, this *"hidden"* information can have negative ramifications on network management decisions regarding traffic engineering, accounting or intrusion detection processes.

REFERENCES

[1] Ben-Basat et al. 2016. Heavy hitters in streams and sliding windows. In *INFOCOM*. IEEE.
[2] Bianchi et al. 2011. On-demand Time-decaying Bloom Filters for Telemarketer Detection. In *CCR, Vol: 41, Num: 5*. ACM.
[3] equinix-chicago dirA. 2018. CAIDA traces. (May 2018). http://www.caida.org/data/passive/passive_dataset.xml.
[4] Liu et al. 2016. One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon. In *SIGCOMM*. ACM.
[5] Sivaraman et al. 2017. Heavy-Hitter Detection Entirely in the Data Plane. In *SOSR*. ACM.