Network Traffic Analysis Using Principal Component Graphs

Harsha Sai Thota HARSHA.SAI@IITG.ERNET.IN
Vijaya Saradhi Vedula SARADHI@IITG.ERNET.IN
T. Venkatesh T.VENKAT@IITG.ERNET.IN
Indian Institute Of Technology Guwahati, Assam, 781039.

Abstract

Traffic dispersion graph (TDG) based methods for traffic classification gained attention due to their visualization power together with inference capability. In this work, we identify the shortcomings in the TDG construction; obtain a generic TDG set and employ principal component analysis on thus constructed set alleviate the identified problems for better classification and analysis of network traffic.

1. Introduction

Accurate classification and analysis of network traffic is important for studying the trends of different applications and their resource usage. Classification of network traffic is a challenging task due to the presence of large number of applications with widely varying characteristics. Network traffic classification methods can be broadly classified into port number based, packet based, flow based, host based and graph based methods. In the **Graph Based** method, network traffic data is represented using a graph known as traffic dispersion graph (TDG) in which IP addresses are represented as nodes and the interaction between any two IP addresses represent an edge. Graph metrics on TDGs help classify application level traffic (Iliofotou et al., 2007). It is demonstrated that application traffic can be identified based on different graph metrics. TDG-based techniques received attention due to their visualization and inference capabilities. However, TDG construction faces several challenges, namely (i) it is not easy to determine the period of trace collection for constructing a TDG; Graph metrics are sensitive to the time intervals and the classification accuracy may get affected (ii) hour of the day during which the trace is collected also affects the constructed TDG (iii) characteristics of network traffic depend on the capacity of the link and the location of trace collection (at the edge or core of the network) which in turn affects the constructed TDG. In the present work, we address all the above shortcomings in TDG construction by constructing a set of TDGs independent of (i) observation time interval (ii) hour of the day traffic is collected and (iii) the trace collection point. A unified set of TDGs is obtained by including all these variations in the TDGs constructed. The input to the PCA is a set of graphs (TDGs) and the output is a set of eigenvalues (variation of bytes within a graph and across the edges of TDGs) and eigenvectors (termed as PCGs) associated with the input data. Each edge in the PCG is then classified into potential application it carries. The entire PCG is then classified according to which application the majority of the edges carry.

2. Traffic Classification using Multiple TDGs through PCGs

We consider two widely used traces, namely (i) trans-Pacific 150 Mbps line (WIDE) collected on different dates[1] and (ii) CAIDA trace collected from 1 Gbps commercial backbone link (equinix-chicago and equinix-sanjose)[2]. Each unique IP address is represented by a node in the TDG. Two nodes are connected by an edge if the corresponding hosts exchange packets in a given time interval on a specific port number. In our experimentation, three port numbers, 53 (DNS), 80 (HTTP) and 443 (HTTPS) are considered. The total number of bytes transferred in the interval is used as weight on the edge. In our experiments, each trace is divided into 4 time intervals. Three TDGs (corresponding to three port numbers) are constructed for each of the 4 time intervals. For example, 900 seconds WIDE trace is divided into 4 equal time intervals (225 seconds) and three TDGs are constructed for the flows observed in 225 seconds. We also constructed 4 TDGs

Appearing in Proceedings of the 1st Indian Workshop on Machine Learning, IIT Kanpur, India, 2013.

[1] http://mawi.wide.ad.jp/
[2] http://www.caida.org

with varying time intervals (150 seconds, 200 seconds, 250 seconds and 300 seconds) making the TDG construction independent of observed time interval and hour of the day. To avoid the influence of trace collection point on the constructed TDGs, we constructed three TDGs from WIDE data sets (on different days of 2009, and 2010) and one TDG from CAIDA data set. We thus have 12 TDGs (4 (time intervals) $\times$ 3 (port numbers)) each capturing the effect of varying time intervals, hour of the day, and trace collection points. Vectorized form of the adjacency matrix representation of a TDG is considered as a single dimension. Principal component analysis is applied on the vector form of the TDGs to obtain principal component graphs. Note that each dimension stands for a TDG. After applying PCA, we obtain as many principal components as the number of dimensions). Each principal axis is termed as a PCG which gives a unified view of the set of TDGs constructed. Obtained PCGs capture the variation of the packet sizes within the edges of the graph and across time intervals.

PCG Edge Classification: Every edge say (i, j) of the PCG is classified as belonging to an application by examining the contribution of the real value of the PCGs and the values are compared with the corresponding packet sizes on edge (i, j) of the set of TDGs. The TDG having close packet size is picked as the match and the corresponding port number (in turn application) is assigned to the PCG edge (i, j). All the edges of every PCG are classified in this manner. PCGs are then termed as pertaining to a particular application if majority of the edges belong to a given application. Figure 1 show experimental results obtained by considering two traffic applications in 4 time intervals ((80, 53) - red, (80, 443) - green and (443, 53) - cyan) with varying time interval for TDG construction. In this figure red bar stand for classifying the 8 obtained PCGs and classifying each edge of the PCG as HTTP or DNS according to the above described method. After classifying each of the edge, majority of the edges in each of the PCG have a pure application traffic, namely HTTP and DNS. The last PCG (PCG8) has least variation and the traffic corresponding to this PCG is noted to be DNS application traffic. Green bar stand for HTTP and HTTPS application traffic. Even though these two applications are linearly not separable, through the proposed PCG based classification method we could achieve as classification accuracy as high as 99.46% across different time intervals using same traffic trace and 100.00% when traffic traces are different (refer to figure 2. Cyan bar represents HTTPS and DNS application traffic classification results. In this case as well we have obtained classification accuracy as high as 98.86% and 100.00% (refer to figure 2) for data using different time intervals and data using different traces. Figure 2 shows experimental results obtained by considering different trace collection points.

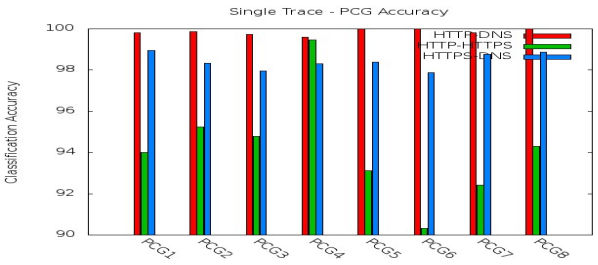

Figure 1. PCG Traffic Analysis: Single Trace, Vary-ing Time Intervals (TDGs - different observed time intervals).

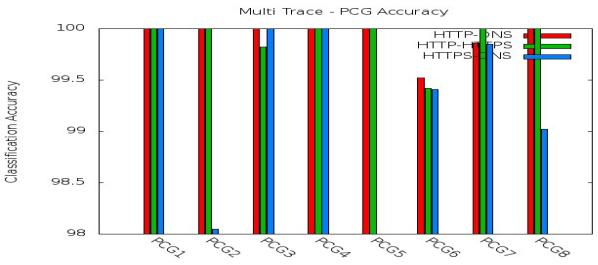

Figure 2. PCG Traffic Analysis: Multiple Traces,Same Time Intervals (TDGs - different trace collection points).

3. Conclusion

Conclusion: We identified issues with the TDG construction and address them using PCGs. PCGs are experimentally shown to be capturing a unified view of the variation of the traffic properties across time and location.

Acknowledgments

We thank all the reviewers of IWML for their valuable comments and suggestions.

References

Iliofotou, Marios, Pappu, Prashanth, Faloutsos, Michalis, Mitzenmacher, Michael, Sing, Sumeet, and Varghese, George. Network monitoring using traffic dispersion graphs (tdgs). In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.