On the Complexity of Traffic Traces and Implications

Chen Avin
School of Electrical and Computer Engineering,
Ben Gurion University, Israel

Manya Ghobadi
Computer Science and Artificial Intelligence Laboratory,
MIT, USA

Chen Griner
School of Electrical and Computer Engineering,
Ben Gurion University, Israel

Stefan Schmid
Faculty of Computer Science,
University of Vienna, Austria

ABSTRACT

This paper presents a systematic approach to identify and quantify the types of structures featured by packet traces in communication networks. Our approach leverages an information-theoretic methodology, based on iterative randomization and compression of the packet trace, which allows us to systematically remove and measure dimensions of structure in the trace. In particular, we introduce the notion of *trace complexity* which approximates the entropy rate of a packet trace. Considering several real-world traces, we show that trace complexity can provide unique insights into the characteristics of various applications. Based on our approach, we also propose a traffic generator model able to produce a synthetic trace that matches the complexity levels of its corresponding real-world trace. Using a case study in the context of datacenters, we show that insights into the structure of packet traces can lead to improved demand-aware network designs: datacenter topologies that are optimized for specific traffic patterns.

CCS CONCEPTS

• Networks → Network performance evaluation; **Network algorithms**; **Data center networks**; **• Mathematics of computing** → *Information theory*.

KEYWORDS

trace complexity, self-adjusting networks, entropy rate, compress, complexity map, data centers

ACM Reference Format:
Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid. 2020. On the Complexity of Traffic Traces and Implications . In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts), June 8–12, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3393691.3394205

1 INTRODUCTION

Packet traces collected from networking applications, such as datacenter traffic, have been shown to feature much *structure*: datacenter traffic matrices are sparse and skewed, exhibit locality, and are bursty. In other words, packet traces from real world applications are far from arbitrary or random.

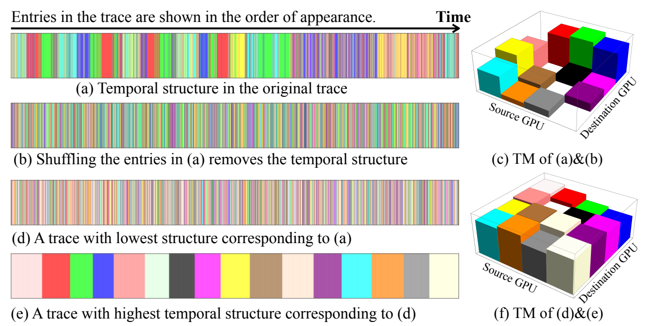

Figure 1: Visualization of temporal and non-temporal structure in a machine learning workload.

However, the available structure can differ significantly across applications, and we currently lack a unified approach to measure the structure in traffic traces in a systematic manner, accounting for both *non-temporal* structures (e.g., how skewed the traffic matrices are) and *temporal* structures (e.g., how bursty traffic is). The quantification of trace structures and their locality can be very useful; it can shed light on the potential of traffic-aware optimization, and facilitate traffic modeling, benchmarking, and synthesis; these are otherwise difficult to achieve, given the limited amount of traffic data available to researchers today.

Let us illustrate the temporal and non-temporal structures available in a traffic trace with an example. Consider a packet trace from a Machine Learning (ML) application based on a popular convolutional neural network training job, with four GPUs. Figure 1(a) visualizes the trace where each packet in the trace is represented by a unique color corresponding to its (source, destination)-GPU pair. The figure highlights the *temporal* structure of the trace: the sequence of colors is far from random. Rather, a pattern is revealed, where certain colors are more frequent in some intervals than in others. For comparison, Figure 1(b) shows the same trace, but randomizes the order of the entries in the trace: the randomization removes the temporal structure observed in Figure 1(a). Intuitively, the trace in Figure 1(a) has more temporal structure than the trace in Figure 1(b). However, the frequency distribution of the two traces is the same: summing up the entries over the entire trace file results in the same traffic matrix (TM), shown in Figure 1(c).

The resulting traffic matrix shows structure as well. In this case the traffic matrix is *skewed*, i.e., some GPU pairs communicate more

SIGMETRICS '20 Abstracts, June 8–12, 2020, Boston, MA, USA

ACM ISBN 978-1-4503-7985-4/20/06.
https://doi.org/10.1145/3393691.3394205

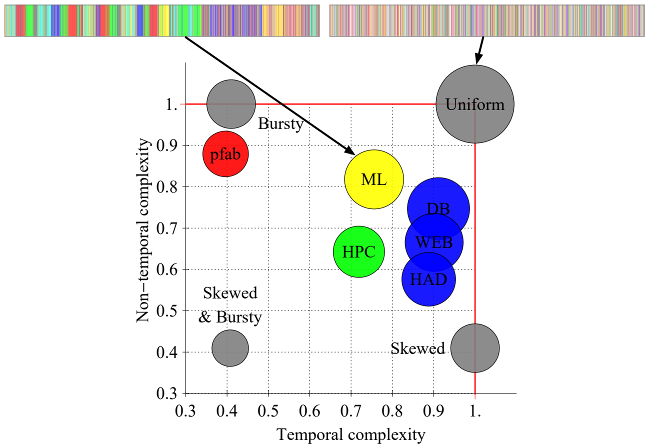

Figure 2: The complexity map of six real traces (colored circles) and four reference points (grey circles at the corners). HAD, WEB, and DB refer to Facebook's Hadoop, web, and database traces [4]. Two corresponding traces from Figure 1, traces (a) and (d), are shown above the map.

frequently than others. That is, the traffic matrix loses information about temporal differences, and it features a *non-temporal* structure. For comparative purposes, consider two synthetic traces shown in Figures 1(d) and (e). Trace (d) is generated *uniformly* and random and has the least temporal structure compared to (a), while trace (e) is *bursty* and built from consecutive source-destination requests, and hence has the most temporal structure. Similarly, the traffic matrix in Figure 1(f) captures the non-temporal structure in both (d) and (e), but not the temporal structure. The traffic matrix is almost uniform, and hence has less structure than (c).

While the different temporal and non-temporal structures in the above traces are obvious and intuitive, we currently lack a systematic approach to measure and quantify them.

This paper takes the first steps to close this gap. In particular, we propose an approach to quantify the amount of temporal and non-temporal structure in traffic traces using the information theory's measure of *entropy*. Since the term entropy is defined for random variables, as opposed to a sequence of individual communication requests in a packet trace, in this paper, we will use the more general term "complexity" to quantify the structure in a packet trace. In particular, we will refer to the complexity of a trace as the *trace complexity*. We will also provide a traffic generation model to produce synthetic traces that match the complexity of a given real world trace. Intuitively, a packet trace with *low entropy* has *low complexity*: it contains little information, and the sequence behavior is more predictable; hence we say that it has *high structure*. Our goal is to enable a unified mechanism to compare the structure pattern in traces, irrespective of the number of nodes and the exact packet arrival times. While prior work focused on providing distributions for flow (or packet) inter-arrival times and sizes [3], we intentionally replace the packet arrival times with the order of arrival to introduce a degree-of-freedom that enables us to compare traces captured in widely different settings.

Our approach allows us to chart, what we call, a *complexity map* of individual traffic traces. More specifically, we map each traffic trace to a two-dimensional graph indicating the amount of temporal and non-temporal complexity that is present in a trace. Figure 2 shows an example of this map. While details will follow later, the map allows us to locate different workloads according to their temporal complexity (x-axis) and their non-temporal complexity (y-axis). The size of the circle represents the total complexity, both temporal and non-temporal. A *uniform* trace without any temporal nor non-temporal structure (like the trace in Figure 1(d) that is shown again above the map) will be located on the upper right corner $x = 1$ and $y = 1$. A trace that is both *skewed* and *bursty* has both temporal and non-temporal complexity that are significantly lower than a uniform trace. The trace of our ML example in Figure 1(a) (shown again above the map) is such an example and is denoted on the map by the yellow circle. A *skewed* trace (like the trace in Figure 1(b)) does not have any temporal structure, so its temporal complexity is maximal: the trace will be located at $x = 1$. However, as this trace contains non-temporal structure, i.e., is skewed (recall the matrix in Figure 1(c)), its y-value may be lower. Given this intuition, we indicate in the figure five additional workloads: three Facebook datacenter workloads (DB, WEB, HAD), a high-performance computing workload (HPC), and a synthetic pFabric [1] workload (pFab). They all provide different complexities, as our approach highlights. We provide more details on these workloads and on the complexity map in the full paper [2].

The main contribution of this paper is a systematic information-theoretic approach to identify and quantify the types of structures (e.g., temporal and non-temporal) featured by packet traces in communication networks. Our approach uses iterative randomization and compression of the packet trace: we iteratively remove and measure dimensions of structure in the trace. We demonstrate an application of our approach in a case study: the design of demand-aware datacenter topologies. In particular, we show that insights into the structure of packet traces can lead to improved network designs that are optimized toward specific traffic patterns. We further present a simple yet powerful model to generate traffic traces that match the complexity of production-level traces, also allowing us to derive theoretical properties of the complexity.

ACKNOWLEDGMENTS

Research supported through ERC Consolidator Grant 864228. Full version appears as [2].

REFERENCES

[1] Mohammad Alizadeh, Shuang Yang, Milad Sharif, Sachin Katti, Nick McKeown, Balaji Prabhakar, and Scott Shenker. 2013. pfabric: Minimal near-optimal datacenter transport. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 435–446.

[2] Chen Avin, Manya Ghobadi, Chen Griner, and Stefan Schmid. 2020. On the Complexity of Traffic Traces and Implications. In *Proc. ACM SIGMETRICS.*

[3] Theophilus Benson, Aditya Akella, and David A Maltz. 2010. Network Traffic Characteristics of Data Centers in the Wild. In *Proc. ACM SIGCOMM Conference on Internet Measurement (IMC).* ACM, 267–280.

[4] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C Snoeren. 2015. Inside the social network's (datacenter) network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication.* 123–137.