Computers and Electrical Engineering 000 (2017) 1–19

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A learning-based measurement framework for traffic matrix inference in software defined networks☆

Mehdi Malboubi[a,*], Shu-Ming Peng[a], Puneet Sharma[b], Chen-Nee Chuah[a]

[a] Department of Electrical and Computer Engineering, University of California, Davis, CA, USA
[b] Networked Systems Group at Hewlett Packard Labs, Palo Alto, CA, USA

ARTICLE INFO

Article history:
Received 24 November 2016
Revised 16 November 2017
Accepted 16 November 2017
Available online xxx

Keywords:
Network measurement and inference
Traffic matrix estimation
Software defined networking
Compressed sensing
Multi-armed bandit algorithms

ABSTRACT

In this paper, we propose an intelligent framework for Traffic Matrix (TM) inference in Software Defined Networks (SDN) where the Ternary Content Addressable Memory (TCAM) entries of switches are partitioned into two parts to: 1) effectively aggregate part of incoming flows for aggregate measurements, and 2) de-aggregate and directly measure the most informative flows for per-flow measurements. These measurements are then processed to effectively estimate the size of network flows. Under hard resource constraints of limited TCAM sizes, we show how to design the optimal and efficient-compressed flow aggregation matrices. We propose an optimal Multi-Armed Bandit (MAB) based algorithm to adaptively measure the most rewarding flows. We evaluate the performance of our framework using real traffic traces from different network environments and by considering two main applications: TM estimation and Heavy Hitter (HH) detection. Moreover, we have implemented a prototype of our framework in Mininet to demonstrate its effectiveness.

1. Introduction

In a network, a flow is a sequence of packets that shares common fields that can be readily extracted from packet headers, such as source and destination IP addresses, port numbers and protocol. The size of an Origin-Destination Flow (ODF) quantifies the volume of traffic between the source node and the destination node (in packets or bytes), and the traffic matrix represents the volume of network flows between all nodes. Fine-grained estimates of traffic matrices provide essential information that is central in many networking applications including network design, network traffic monitoring and management [1,2], and network anomaly detection and security [3]. There are two main approaches for TM measurement and estimation. Direct flow-based measurements such as NetFlow and Sampled Flow (sFlow) offer fine-grained measurements that can support different measurement tasks. However, such an approach not only requires dedicated hardware and specialized algorithms, but it is also often challenging, inefficient or even infeasible to monitor each and every flow due to exploding traffic volume and limited monitoring resources (e.g., the number of TCAM entries, storage capacity and processing power). Accordingly and due to non-scalability of direct flow measurement techniques in large-scale networks, intelligent sampling and streaming algorithms have been proposed to estimate statistics or answer specific queries of, e.g.,

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Zhihan Lu.

* Corresponding author.
E-mail addresses: mmalboubi@ucdavis.edu (M. Malboubi), shmpeng@ucdavis.edu (S.-M. Peng), puneet.sharma@hpe.com (P. Sharma), chuah@ucdavis.edu (C.-N. Chuah).

https://doi.org/10.1016/j.compeleceng.2017.11.020

approximate size of elephant flows [4]. These solutions are task-specific and lack the full flexibility to dynamically choose which traffic sub-population to measure, and how, depending on application requirements.

An alternate approach is estimating the traffic matrix based on a limited set of measurements using Network Inference (NI) and network tomography methods. However, most network inference problems are naturally formulated as ill-posed Under-Determined Linear Inverse (UDLI) problems where the number of measurements are not sufficient to uniquely and accurately determine the solution. Hence, side information from different sources must be incorporated into the problem formulation to improve estimation precision [5,6].

In today's large-scale networks, the flows of interest can be sparse, or highly fluctuating over time and/or space, due to the presence of a variety of applications and services with different behaviors and Quality of Services (QoS) requirements. The management and operation of today's networks, and providing high QoS for a variety of applications in these complex and highly dynamic networks, depend on the accurate measurement and estimation of the size of network flows. In fact, many recent network operation, monitoring and security applications require timely estimates of both large and small traffic flows with high accuracy. Therefore, the network measurement infrastructure must be agile enough to cope with the dynamic network and traffic conditions. Such a flexible architecture can be achieved due to the recent advent of software defined networking. In fact, an SDN enabler, such as OpenFlow, nicely separates the measurement data plane and control plane functions, and provides the capability to control/re-program the internal configurations of switches in dynamic environments. Consequently, SDN allows applying more complex network monitoring and management applications which results in providing more effective and efficient network services [7,8].

An important resource in SDN switches and routers is TCAM entries which are essential for effective flow classification and forwarding in today's high-speed networks. TCAMs are special computer memory with the capability of real-time and exact-match search which can be used for noncontiguous and arbitrary bitwise packet matching and classifications in network switches and routers [9]. However, TCAMs are expensive hardware and consume a lot of power; hence, the number of TCAM entries in network switches and routers are limited. Therefore, to efficiently allocate such an important resource in dynamic environments, where the behaviors of operating network change, effective optimization and adaptive learning algorithms must be utilized. In fact, adaptive learning algorithms and optimization techniques give us the capability of learning the behavior of network flows and optimally allocating TCAM resources among different network operation and monitoring applications. For this purpose, the run-time programmability of SDN plays an important role to achieve such an intelligent framework with the ability of reconfiguring measurement modules and constructing effective TCAM rules, and to adaptively and easily collect a set of timely-effective flow statistics using counters associate with TCAM entries [10–13].

Accordingly, we propose a framework called intelligent SDN based Traffic (de)Aggregation and Measurement Paradigm (iSTAMP). In iSTAMP, measurement modules can be configured on-the-fly to collect fine-grained measurements of specific traffic sub-populations of interest that directly reflect the monitoring application requirements. Based on the philosophy that not all attributes of interest are equally important, iSTAMP utilizes an intelligent sampling algorithm to select the most informative traffic flows, using information gathered throughout the measurement process. It also exploits Compressed Sensing (CS) inference methods that are effective for estimating highly fluctuated sparse unknown quantities from a set of well-defined compressed linear measurements [14,15].

In order to do this, iSTAMP leverages OpenFlow to dynamically partition the TCAM entries of a switch/router into two parts. In the first part, a set of incoming flows are optimally aggregated to provide well-compressed aggregated flow measurements that can lead to the best possible estimation accuracy via network inference process. The second portion of TCAM entries are dedicated to track/measure the most *rewarding* flows (defined as flows with the highest impact on the ultimate monitoring application performance) to provide accurate per-flow measurements. These flows are selected and "stamped" as important (or *rewarding* from monitoring's perspective) using an intelligent multi-armed bandit based algorithm. These two sets of measurements (aggregated and sampled flows) are then jointly processed to estimate the size of all network flows using different optimization techniques.

Therefore, iSTAMP is scalable framework because it uses a limited number of aggregated flow measurements to estimate the size of a large number of flows. In addition, iSTAMP produces accurate flow estimates since: 1) it uses compressed sensing NI techniques which are effective for estimating highly fluctuated network flows, and 2) part of the flows are perfectly and directly measured by intelligently tracking them, depending on the application.

1.1. Related work

There is a rich literature on improving the accuracy of traffic flow measurements and estimation. We will briefly discuss the most relevant work here. In [16], ProgMe proposes a re-programmable architecture allowing statistics collection based on the notion of flowsets (arbitrary traffic sub-populations) using a flexible flowset composition language. From SDN perspective, most current research has focused on leveraging the run-time programmability/reconfigurability of SDN to passively measure the size of the sub-set of network traffic flows, or identify heavy hitters and/or Hierarchical Heavy Hitters (HHH), which is of a particular importance in different traffic monitoring and network security applications. In [7,8,17], SDN reconfigurable measurement architectures are proposed where a variety of sketches for different direct measurement tasks can be defined and installed by the operator. In [18], an adaptive flow counting method is proposed that controls the temporal and spatial aggregation using a linear prediction method. In addition, in [19], OpenTM directly measures the network's TM by keeping track of statistics for each flow. However, these methods suffer from two challenges: first, the pure SDN