Software defined Network Inference with Passive/active Evolutionary-optimal pRobing (SNIPER)

***Abstract*—A key requirement for network management is accurate and reliable network monitoring where critical information about internal characteristics or states of the network(s) must be obtained. In today's large-scale networks, this is a challenging task due to the hard constraints of network measurement resources. In this paper, a new framework (called SNIPER) is proposed where we use the flexibility provided by Software-Defined Networking (SDN) to design the optimal observation or measurement matrix which leads to the best achievable estimation accuracy using Matrix Completion (MC) techniques. Here, to cope with the inherent complexity of the process of designing large-scale optimal observation matrices, we use the well known Evolutionary Optimization Algorithms (EOA) which directly target the ultimate estimation accuracy as the optimization objective function. We evaluate the performance of SNIPER using both synthetic and real network measurement traces from different network topologies and by considering two main applications including network traffic and delay estimations. Our results show that this framework is generic and efficient that can be used for a variety of network performance measurements under hard constraints of network measurement resources. Also, to demonstrate the effectiveness and feasibility of our framework, we have implemented a prototype of SNIPER in Mininet.**

I. Introduction

Software-Defined Networking (SDN), along with its OpenFlow enabler, ~~is an emerging technology that~~ nicely separates the ~~measurement~~ data plane and control plane functions, and provides a capability to control and re-program the internal configurations of switches in dynamic environments. Such a flexibility can be used to adaptively and efficiently implement more complex networking applications, including both passive and active network monitoring without the need of ~~customization~~. This is of particular importance in many network management and security applications where accurate and reliable network monitoring is necessary to provide critical information about internal characteristics or states of the network(s) that must be directly measured or indirectly inferred [1] [2] [3] [4].

In today's complex networks, the direct measurement of network's Internal Attributes of Interest (IAI) can be challenging or even ~~inefficient and~~ infeasible due to the hard constraints of network measurement resources. For example, consider the case where the IAI is the Origin-Destination Flow (ODF) sizes (as a measure of traffic intensity between nodes) or where the IAI are per-flow delay, throughput or packet loss. In large-scale networks, the measurement resources, including the Ternary Content Addressable Memory (TCAM) entries, processing power, storage capacity and limited available bandwidth, are very limited and ~~providing~~ per-flow measurements ~~are not possible~~. ~~Accordingly,~~ Network Inference (NI) techniques~~, as powerful network monitoring tools,~~ can be ~~used~~ to estimate the IAI based on a limited set of passive and/or active measurements. However, NI problems are naturally ill-posed in the sense that the number of measurements are not sufficient to uniquely and accurately determine the solution. Hence, side information from different perspectives and sources must be incorporated into the problem formulation to improve the estimation precision [5] [6] [7].

The flexibility provided by the SDN can be utilized to ~~optimize and~~ facilitate the process of ~~providing~~ required direct measurements and/or side information which can then be used by network inference techniques to estimate the IAI. ~~Accordingly,~~ the capabilities of SDN have been utilized in a variety of passive and active network monitoring applications. ~~For passive network measurement, the~~ most SDN based works related to traffic engineering and network security applications ~~where the main goal is focused on network traffic measurement or~~ identifying Heavy Hitters (HH) and Hierarchical Heavy Hitters (HHH). In [1] and [2], 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 [8], OpenTM directly measures a traffic matrix by keeping track of statistics for each flow. Recently, in [3], an intelligent SDN based traffic measurement framework (called iSTAMP) with the ability of adaptive and accurate fine-grained flow estimation is proposed. For active network measurement under SDN paradigm, the very recent work [4] establishes a general framework (called Opennetmon) where accurate measurements of per-flow throughput, packet loss and delay can be directly measured.

However, under ~~hard constraint of network measurement resources where network inference methods must be used to estimate the IAI from a set of limited measurements~~, the state of the art method ~~in~~ [3] suffers from the following challenges. First, the application of this framework is mainly limited to network traffic measurement. Second, ~~the limited capability of current SDN infrastructures for providing required side information (due to~~ longest prefix matching forwarding scheme ~~in which~~ incoming flows can be aggregated in just one entry of the TCAM~~)~~ ~~hardly~~ limits the capability of providing optimal redundant aggregated measurements. Third, in [3], to simplify the process of designing the optimal aggregation (i.e. measurement) matrix, the coherency of the measurement matrix is minimized ~~which does not target~~ the main estimation accuracy, ~~and therefore, the~~ unavoidable sacrifice in the performance ~~is accepted~~ [3][9].

On the other hand, recently, Matrix Completion (MC) techniques have been used as powerful network inference tools ~~where the problem is the completion of~~ a matrix of IAI from the direct measurement of a sub-set of its independent entries [10][11][12]. Examples of the matrix of IAI include a matrix where each entry is an ODF at different times, or per-flow delay/packet-loss between different nodes of the network. Since ~~in communication networks a variety of resources of the communication infrastructure at~~ different layers ~~are shared~~, the main assumption in MC techniques is that the matrix of IAI is a low-rank matrix which contains spatio-temporal redundancies and thus not all of its entries are needed to represent it; accordingly missed or non-observed entries can be estimated from a sub-set of randomly measured entries. In the theory of matrix completion, the matrix of IAI can be completely reconstructed from a sub-set of observed/measured entries (indicating the observation/measurement matrix) if the number of *randomly* chosen observations are *high* enough [13][14]. Accordingly, in [10] and [11] the MC methods are used for network Traffic Matrix (TM) completion to estimate the missed entries of the TMs. Also, in [12], a new MC technique has been used for active network performance measurements where the status of path delays or bandwidths are predicted from a set of active measurements and using a new MC technique.

~~The construction of the~~ matrix of IAI, where its entries are the measurements of *independent* IAI, ~~along~~ with the flexibility provided by the SDN pave the way ~~to~~: 1) ~~design~~ an efficient framework for different passive/active network measurement applications under hard constraint of measurement resources, and 2) ~~provide~~ required side information without feasibility constraints ~~(as in [3])~~. ~~Accordingly, in~~ this paper, we use different matrix completion techniques as our main NI tools, and define an observation matrix as a matrix which provides required direct measurements of IAI that can be used by MC algorithms, ~~and consequently,~~ we answer the following interesting question:

Under hard ~~resource constraints of network measurement resources~~, how can we optimally measure or sample a sub-set of entries of the matrix of IAI and design the optimal observation matrix which leads to the best possible estimation accuracy ~~via~~ using matrix completion techniques?

However, the *direct* design of optimal observation matrices for maximizing the performance of NI methods is prohibitive due to the complexity of the process [3][9]. The underlying difficulty ~~in this direct optimal observation matrix design, where the ultimate estimation accuracy of the NI process is targeted, is~~ the fact that formulating the network inference process or algorithm into a closed-form and well-defined mathematical objective function ~~that can be efficiently optimized~~ is extremely complicated and computationally complex, if it is not impossible or intractable. Therefore, in this paper, we propose a new approach in designing the optimal observation matrix for network inference problems where we *directly* target the ultimate estimation accuracy in network monitoring applications in our optimization framework. However, to cope with the inherent complexity of the process of designing large-scale optimal observation matrices, we use the well known Evolutionary Optimization Algorithms (EOA) that are suitable for the optimization problems where the main objective function is a procedure or an algorithm that can not be formulated as a well-defined mathematical function. In this framework, the evolutionary optimization algorithm acts as a *sniper* which precisely captures or measures the best or the most informative entries of the matrix of IAI which leads to the best estimation accuracy via using matrix completion techniques.

~~Under hard constraints of measurement resources in network monitoring applications, the~~ SNIPER is a simple, flexible, and efficient framework which can be easily deployed on commodity OpenFlow-enabled routers/switches to enhance the performance of various passive or active network monitoring applications with low computation and communication overhead between control and data planes. Since MC techniques ~~are the~~ ~~main NI methods in SNIPER, which~~ directly use partial independent measurements of IAI, ~~thus,~~ this framework ~~not only can be easily implemented in a centralized manner but also it can be implemented in distributed or decentralized ways~~. Accordingly, it is compatible with the recent trends in developing more smart and agile SDN platforms [15][16] where data plane APIs and switches are able to execute codes inside the device with no further interaction with the controller.

Our main contributions are summarized as follows:

- To the best of our knowledge ~~for the first time, we use the~~ EOAs to design the optimal observation matrix where ultimate network inference performance is the main objective function to be optimized. We show that under hard ~~constraint of measurement resources~~ the optimal design of the observation matrix provides more accurate estimates in different network monitoring applications, including per-flow size and delay estimations.
- We address the scalability, deployability and feasibility of the SNIPER framework: a) by reducing the computational complexity of EOAs; b) by introducing a new algorithm for network measurement in dynamic environments, and c) by evaluating the performance of our framework using both synthetic and real network measurement traces from different network topologies ~~in~~ two main applications ~~including~~ network traffic and delay estimation, ~~and further, by implementing~~ a prototype of SNIPER in Mininet environment.

The rest of this paper is organized as follows. Section II provides an overview of the SNIPER framework and the matrix completion techniques that we have used as our main NI methods. In Section III we describes our optimal observation matrix design procedure using the EOAs. Then, in Section IV, we explain our methodology for evaluating the performance of the SNIPER. ~~Accordingly,~~ in Section V, we ~~evaluate the performance~~ of SNIPER considering two main applications including per-flow path delay and per-flow size estimations. Section VI summarizes our most important results.

II. SNIPER: System Description

Figure 1 shows the general block diagram of the SNIPER framework where the controller interacts with Software Defined Measurement Network (SDMN) via Network Management (NM) and Optimal Network Measurement Probing (ONMP) messages to reconfigure the SDMN and poll the required measurements and statistics of the operating network. The SDMN consists of a sub-set of OpenFlow Switches (OFS) in the operating network which guarantees all required IAI are *observable* and *measurable* using the SDMN, ~~as it is our assumption here~~. The NM messages include regular

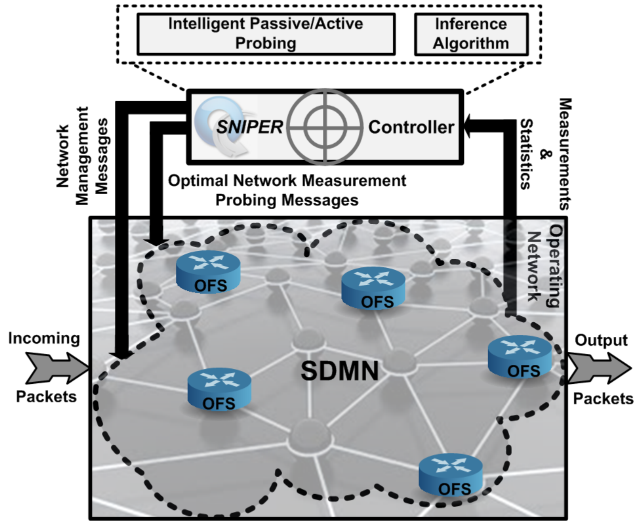
Intelligent Passive/Active Probing
Inference Algorithm
SNIPER
Controller
Network Management Messages
Measurements & Statistics
Optimal Network Measurement Probing Messages
Operating Network
OFS
OFS
OFS
OFS
OFS
SDMN
Incoming
Packets
Output
Packets

Fig. 1: SNIPER network measurement framework: a general perspective.

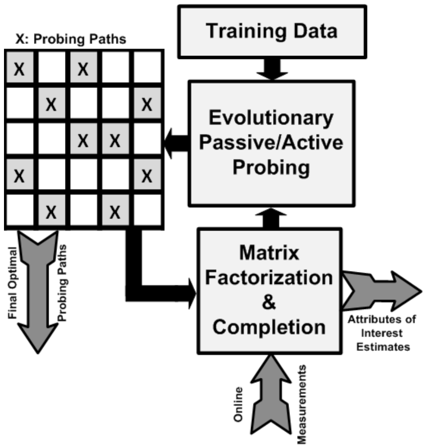
X: Probing Paths
Training Data
Evolutionary Passive/Active Probing
Matrix Factorization & Completion
Attributes of Interest Estimates
Final Optimal Probing Paths
Online Measurements

Fig. 2: Evolutionary optimal network probing process.

network operating and control commands while ONMP messages include passive/active probing messages ~~which exactly~~ indicate which IAI must be accurately measured at different times and/or spaces. In the SNIPER framework, the network measurement process is consisted of two stages, ~~including~~ learning and measurement stages, as ~~it is~~ shown in Figure 2. In the learning stage, the ONMP messages are designed and computed using a supervised learning scheme where an evolutionary optimization algorithm uses a training data set to precisely design the ONMPs. Then, in the measurement stage, the flexibility provided by the SDN is used to reconfigure the SDMN and to collect the measurements of corresponding ONMPs. By decreasing the dependency of the SNIPER framework on the initial training data set and adaptively updating the optimal observation matrix, this framework can also be used in dynamic environments.

The measurements are transmitted to the SNIPER controller where matrix completion techniques are used as the main NI algorithm to estimate all unknown IAI. The SNIPER controller can both preconfigure the switches with forwarding rules as well as ~~it can~~ reactively respond to requests from switches, ~~which are~~ sent when a packet matches none of the existing rules enters the network. Besides managing the forwarding plane, the SNIPER controller is also capable of requesting per-flow counter statistics and injecting packets into the network. Accordingly, the SNIPER architecture can obtain information of ~~running~~ flows and regularly injects ~~packets~~ into the network to monitor end-to-end network performance measurements [3][4].

The ONMPs are consisted of passive and active probes. Passive probes are measurement massages that reconfigure the OFSs of the SDMN by installing different rules in flow tables and defining the required actions on the incoming packets of the operating network. ~~However,~~ active probes not only install the rules of flow table(s) and define their actions but also ~~use part of the network resources to~~ inject pre-defined ~~network~~ measurement ~~packets (probes)~~ which will be ~~captured~~ by reconfigured OFSs and provide corresponding statistics. The feasibility of such Software-Defined Measurements (SDM) have been independently investigated in [3] and [4] where the capability of OpenFlow switches can be effectively utilized to measure and inference the IAI of the operating network. Likewise, the SNIPER controller is capable of providing: 1) per-flow sizes [3] by installing the flow ID prefixes in the flow tables and polls the statistics; 2) per-flow throughput [4] by determining specific path for each flow and queries switches to retrieve per-flow statistics where each query determines the amount of bytes sent and the duration of each flow; 3) per-flow packet loss [4] by polling flow statistics from the first and last switch of each path and subtracting the increase of the source switch packet counter with the increase of the packet counter of the destination switch, and 4) per-flow delay [4] by assigning a specific path to each flow and regularly injecting packets into the first switch and having the last switch send them back to the controller where the difference between the packet's departure and arrival times are computed by subtracting the estimated latency from the switch-to-controller delays.

In this paper, the performance of the SNIPER framework is mainly examined in two main applications ~~including~~ per-flow size and delay estimation both synthetic and real network measurement traces from different network topologies. In addition, a prototype of SNIPER is implemented in the Mininet to demonstrate its ~~implementation's~~ feasibility and effectiveness.

A. SNIPER: Problem Statement

Network monitoring is the problem of inferring the IAI corresponding to the performance of the operating network. The operating network is modeled as a connected undirected graph $G(V, E)$ where $|V| = N$, and $|E| = m$. Accordingly, there are M links, and $n = N(N-1)$ paths and flows in the network, assuming that there exists an unique path between any pair of nodes in the network. As it was explained in the introduction, in the SNIPER framework the NI problem is modeled as a Matrix Completion (MC) problem where the problem is the completion of a matrix of attributes of interest (X) from the direct measurement of a sub-set of its entries assuming that X is a low-rank matrix which contains redundancies and thus not all of its entries are needed to represent it. Here, X is a matrix of size $n \times \mathcal{T}$ ($\mathcal{T} < n$) with rank $r << \mathcal{T}$ where m entries of X is directly measured; the