A Modular Sampling Framework for Flexible Traffic Analysis

João Marco C. Silva, Paulo Carvalho and Solange Rito Lima
Centro Algoritmi, Universidade do Minho, Braga, Portugal
Email: joaomarco@di.uminho.pt, pmc@di.uminho.pt, solange@di.uminho.pt

Abstract—**The paradigm of having everyone and everything connected in an ubiquitous way poses huge challenges to today's networks due to the massive traffic volumes involved. To turn treatable all network tasks requiring traffic analysis, sampling the traffic has become mandatory triggering substantial research in the area. Aiming at fostering the deployment and tuning of new sampling techniques, this paper presents a flexible sampling framework developed following a multilayer design in order to easily set up the characteristics of a sampling technique according to the measurement task to be assisted. The framework implementation relies on a comprehensive sampling taxonomy which identifies the *granularity*, *selection scheme* and *selection trigger* as the inner characteristics distinguishing current sampling proposals. As proof of concept of the versatility of this framework in testing the suitability of distinct sampling schemes, this work provides a comparative performance evaluation of classical and recent sampling techniques regarding the estimation accuracy, the volume of data involved in the sampling process and the computational weight in terms of CPU and memory usage.**

I. Introduction

The heterogeneity and massive traffic volumes crossing todays networks have impelled sampling as mandatory technique to accomplish effective measurement and monitoring tasks.

Although most measurement points (MPs), whether running in a dedicated device or embedded in switches or routers, provide tools following IETF sampling directions in RFC 5475 [1], many recent works have proposed sampling techniques and policies (often not supported in current off-the-shelf network equipment) that achieve better results regarding the accuracy in metrics estimation or the reduction of computational overhead for various measurement tasks. Therefore, by analyzing current sampling techniques through its constituent parts, rather than a closed unit, allows to identify their common properties and address eventual constraints (related to accuracy, data overhead and computational weight) within a narrower and simpler scope.

In this context, this paper presents a modular sampling framework based on a taxonomy of sampling techniques which allows to test and deploy flexible sampling-based measurement systems. The taxonomy was developed with the aim to clarify sampling concepts and to provide a common ground for current and forthcoming sampling proposals. The sampling framework follows a multilayer design, implementing the components identified in the taxonomy as functional modules. The usability and versatility of the framework is here assessed through a comparative study of classical and recently proposed sampling approaches, including the analysis of the underlying tradeoff among estimation accuracy, volume of data involved and computational weight in performing different network activities. In order to provide a conceptual framing, this paper starts by providing a global view of key components to sustain a versatile and lightweight sampling-based measurement strategy, through the proposal of a three-layer measurement architecture and corresponding operation.

The remaining of this paper is organized as follows: the related work is discussed in Section II; the main characteristics of sampled-based measurement systems are introduced in Section III; the sampling framework and taxonomy are detailed in Section IV; their applicability in a comparative study of current sampling proposals is carried out in Section V; conclusions are included in Section VI.

II. Related work

Currently, traffic sampling techniques sustain a broad range of network tasks including: *network management* involving short, medium and long term planning and management of network operation, maintenance and provisioning of network services [2] [3]; *traffic engineering* involving performance optimization, traffic characterization, traffic modeling and control [4] [5]; *performance evaluation* of protocols and management tools, network reliability and fault tolerance [6] [7]; *network security*, including anomalies and intrusion detection, botnet and DDoS (Distributed Denial of Service) identification [8]; *SLA (Service Level Agreement) compliance*, where auditing tools might resort to network sampling for measuring and reporting service levels [9]; *QoS control*, an area widely assisted by sampling, for measuring parameters such as delay, jitter and packet loss [10] [11].

Although many of these proposals achieve better results when compared to the techniques defined in [1], there are some barriers that hamper their usage in large scale. One of them is the lack of some important and standard approaches in representative sampling tools, such as Cisco Sampled NetFlow and sFlow (standardized in RFC3176).

The wide adoption of new sampling approaches can be fostered through a flexible measurement architecture capable of selecting adequate traffic sampling components. Therefore, defining a taxonomy identifying sampling inner components and implementing a modular sampling framework are key aspects toward the former objective.

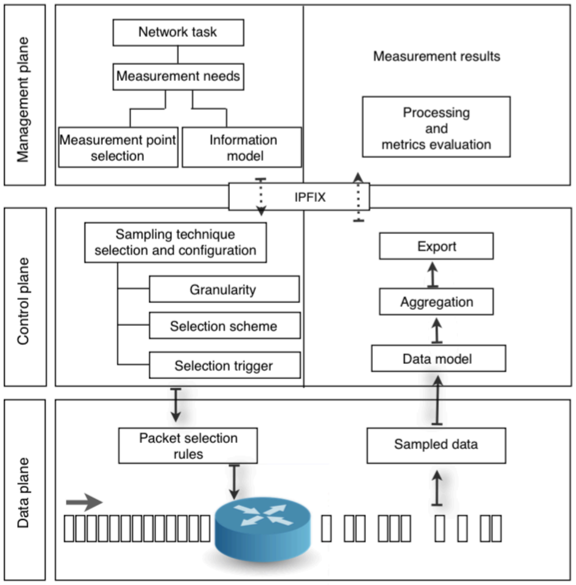


Fig. 1. Sampling-based measurement architecture

III. A SAMPLING-BASED MEASUREMENT ARCHITECTURE

The main components involved in the proposed sampling-based measurement architecture are arranged in three planes, as illustrated in Figure 1. The *management plane* includes tasks deployed directly in measurement points or in external management entities. Based on requirements of each network task, measurement needs are identified and one or more measurement points are selected to participate in the sampling process. This also involves identifying an information model able to define managed objects in the network independently of specific implementations or protocols in use, as suggested in RFC6727. The management plane, apart from providing the corresponding configuration parameters to the control plane, is responsible for estimating the relevant metrics using data reports sent by the control plane. The required processing might involve results from single or multipoint measurements.

A modular design of the *control plane* allows a flexible sampling technique selection and configuration. Considering IETF inputs from the former Packet Sampling Working Group and recent sampling proposals, a sampling taxonomy is here defined to identify the inner characteristics distinguishing sampling techniques (see Section IV-A). The taxonomy also supports the definition of new sampling techniques which can be adjusted to each traffic/service measurement scenario.

In the control plane, the sampled packets received from the data plane are processed and the relevant field contents are extracted according to the network task measurement needs. These values are then aggregated (both in time and space) and exported following IETF IP Flow Information eXport (IPFIX) specifications (*i.e.*, RFC6728).

At data plane, traffic is collected from network interfaces by applying the sample rules defined in the control plane. The unprocessed packets are then reported to the control plane to be processed, simplifying the data plane.

In this measurement architecture, identifying and articulating the sampling process components is a major design issue for achieving an encompassing and efficient sampling solution. In this context, we have implemented a flexible and modular sampling framework by selecting and configuring each sampling technique according to the measurement purpose.

IV. SAMPLING FRAMEWORK

A. The sampling taxonomy

The defined taxonomy fragments the sampling techniques into three well-defined components according to the *granularity*, *selection trigger* and *selection scheme* in use. Then each component is further divided into a set of approaches commonly followed in both classic and recently proposed sampling techniques. Table I details the proposed taxonomy, following the preliminary classification included in [12].

The sampling framework was developed following the taxonomy presented above and the relationship among its components. The framework design can be seen as a multilayer system in which a lower layer provides services to an upper layer, hiding details about its operation. Aiming at multiplatform support and compatibility, the framework was deployed using *libpcap* as the capture interface between the framework core and the network interface being sampled. The data plane is also kept simple, since this library is widely used in important measurement tools, e.g., *tcpdump*. The framework is available for download at http://1drv.ms/1IggkCa as a Raspbian image ready to be deployed.

Regarding the deployment, as presented in Figure 2, the *granularity* component is deployed in a single class and each different approach corresponds to a method. The *flow-level* method receives the *flow key* parameter from which the packets will be sampled, following the *tcpdump* filter syntax. This allows to define flows beyond the classic 5-tuple scheme, extending the concept of flow, as suggested in RFC2724. Considering that in the *packet-level* approach all packets are eligible by the sampling process, its corresponding method does not need to receive parameters, forwarding all packets to the upper layer of the framework, *i.e.*, the *selection trigger*.

Similarly, the *selection trigger* component is also deployed in a single class with specific methods implementing each approach. As presented in Figure 2, the *count-based* method receives two parameters *i.e.*, the *interval between samples* and the *sample size*. These parameters correspond to (i) the number of packets ignored for measurement purposes, therefore not collected, and (ii) the number of packets collected to compose a sample and consequently collected and stored. Each invocation of this method starts a single sample collection, this allows its usage in adaptive techniques, in which the sampling frequency might vary during the measurement process. The *time-based* method receives similar parameters, however they correspond to the timestamps (in milliseconds) of the packets arriving at the measurement point. The *event-based* method receives a string indicating which packet fields must match in order to capture specific packets. This parameter also follows the *tcpdump* filter syntax.

TABLE I
TAXONOMY OF SAMPLING TECHNIQUES

<table>
<tr><th colspan="2">Granularity</th></tr>
<tr><td colspan="2">This component identifies the atomicity of the element under analysis by defining which segment of traffic is considered in the sampling process and in the data reporting format.</td></tr>
<tr><td><i>Flow-level</i></td><td><i>Packet-level</i></td></tr>
<tr><td>The traffic capture policy is only applied to packets belonging to a flow or a set of flows of interest.</td><td>In a first instance, packets are collected indistinctly, as an independent entity, for subsequent filtering or aggregation.</td></tr>
</table>

<table>
<tr><th colspan="3">Selection trigger</th></tr>
<tr><td colspan="3">This component is used to decide the spacial and temporal sample boundaries by defining the start and the end of a sample, and consequently the interval between samples.</td></tr>
<tr><td><i>Count-based</i></td><td><i>Time-based</i></td><td><i>Event-based</i></td></tr>
<tr><td>The beginning and the end of a sample are driven by the spatial position of the packet within the traffic stream, using counters which are independent of the packet arrival time.</td><td>The beginning and the end of a sample is determined based on packet arrival time. When a new sample is triggered, the MP collects all further incoming packets until the end of the sample interval.</td><td>The decision on when a sample starts and ends takes into account some particular event observed in the traffic being monitored, such as some value in the packet contents or the treatment of the packet at the measurement point.</td></tr>
</table>

<table>
<tr><th colspan="3">Selection scheme</th></tr>
<tr><td colspan="3">This component identifies the selection function that determines which packets will be selected and collected.</td></tr>
<tr><td><i>Systematic</i></td><td><i>Random</i></td><td><i>Adaptive</i></td></tr>
<tr><td>The process of packet selection is ruled by a deterministic function which imposes a fixed sampling frequency, independently of the packet content or treatment.</td><td>The sampling frequency is ruled by a probabilistic function [1] that can be uniform, where all packets have an equal probability to be selected, or otherwise, non-uniform.</td><td>The selection process is able to change the packet selection criterion during the course of measurements in response to the traffic behavior, expected accuracy or resource constraints.</td></tr>
</table>

Considering that the *selection scheme* component corresponds to the main distinguishing feature among sampling techniques, involving possibly complex functions, each approach in this component is implemented as a single class. This promotes flexibility when deploying new techniques, as the methods within the selection trigger and granularity components are kept invariable, as presented in Figure 2.

The *systematic* approach, as defined in [1], comprises the simplest sampling techniques and consists in successive invocations of the same method from the selection trigger object using invariable parameters. The *random* approach includes a random generator method which may follow different probabilistic functions. Considering that the portion of the traffic collected vary in every sampling iteration, each invocation of selection trigger method receives different parameters. Note that the sample size does not change, only its temporal or spatial position does. The *adaptive* approach is usually the most complex within the selection scheme component as it requires monitoring of a reference parameter, *e.g.* throughput, that will guide the sampling adaptiveness [13], [14]. It also resorts to a controller designed to analyze the reference parameter in order to decide on the sampling frequency. This is accomplished through specific method invocations from the selection trigger object varying the distribution and/or the sample size.

V. EXPERIMENTAL RESULTS

The developed framework is currently supporting research work related to the suitability of the different sampling techniques when applied to various network measurement activities, taking into account the measurement accuracy, volume of data involved and computational weight. In particular, the experimental tests reported in this work evaluate the impact of sampling techniques on network flow analysis.

1) Traffic scenarios and sampling techniques: The performance analysis carried out resorts to real public traffic traces captured in OC-48 and OC-192 links (available from CAIDA), containing over 13 and 15 millions of packets, respectively.

The sampling techniques under analysis comprehend classical approaches widely deployed in current tools, which are in compliance with [1], and recently proposed approaches. In more detail, the analysis include: SystC - Systematic count-based [1]; SystT - Systematic time-based [1]; RandC - Random count-based (uniform probability) [1]; LP - Adaptive linear prediction (time-based) [13]; and MuST - Multiadaptive (time-based) [14]. The following comparative evaluation uses the frequency 1/100 for SystC and RandC techniques, as suggested in [15]. For SystT technique, the sampling frequency in use is 100/1000. To avoid biasing the analysis by significative differences in the volume of sampled traffic, evaluation tests using SystC 1/32 and SystC 1/16 were also carried out, as these SystC parameters produce an amount of sampled traffic similar to MuST and SystT, respectively.

A. Evaluating the sampling techniques

The sampling techniques are evaluated regarding the trade-off among volume of data involved, estimation accuracy and computational weight in performing traffic flow analysis.

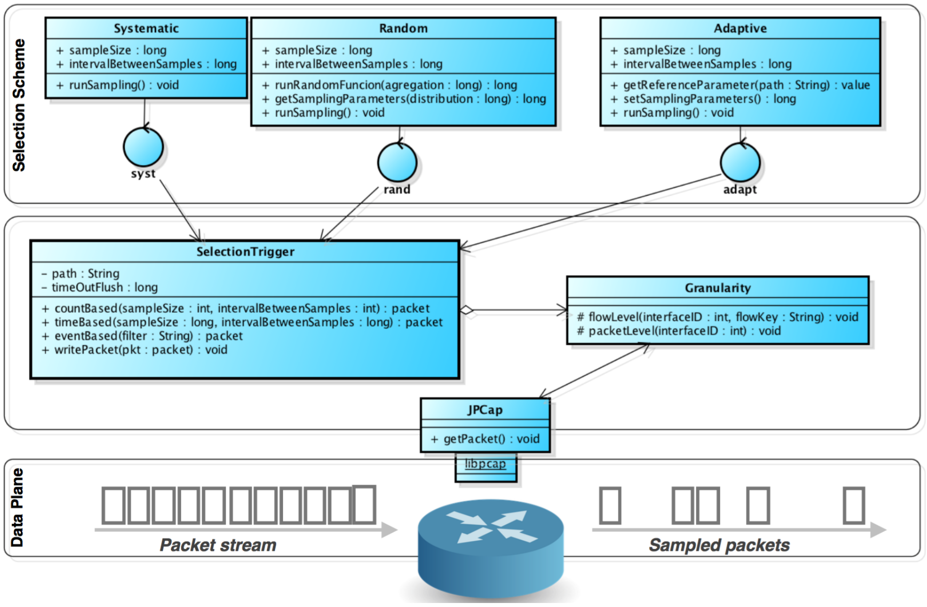

Selection Scheme
Systematic
+ sampleSize : long
+ intervalBetweenSamples : long
+ runSampling() : void
Random
+ sampleSize : long
+ intervalBetweenSamples : long
+ runRandomFuncion(agregation : long) : long
+ getSamplingParameters(distribution : long) : long
+ runSampling() : void
Adaptive
+ sampleSize : long
+ intervalBetweenSamples : long
+ getReferenceParameter(path : String) : value
+ setSamplingParameters() : long
+ runSampling() : void
syst
rand
adapt
SelectionTrigger
- path : String
- timeOutFlush : long
+ countBased(sampleSize : int, intervalBetweenSamples : int) : packet
+ timeBased(sampleSize : long, intervalBetweenSamples : long) : packet
+ eventBased(filter : String) : packet
+ writePacket(pkt : packet) : void
Granularity
# flowLevel(interfaceID : int, flowKey : String) : void
# packetLevel(interfaceID : int) : void
JPCap
+ getPacket() : void
libpcap
Data Plane
Packet stream
Sampled packets

Fig. 2. Main classes in the framework

Regarding the overhead in terms of volume of data (in MBytes and in number of sampled packets facing the total unsampled traffic), Figure 3 shows that, for the sampling frequencies considered, the count-based approaches require less storage and transmission resources. For time-based approaches, the MuST technique presents lower resource requirements. Globally, the sampled traffic represents a very low percentage of the total traffic trace, below 15%. Despite the importance of reducing the consumption of resources associated with traffic analysis, specially when facing today's massive traffic volumes, to be effective, the sampling techniques must still be able to represent the network status accurately.

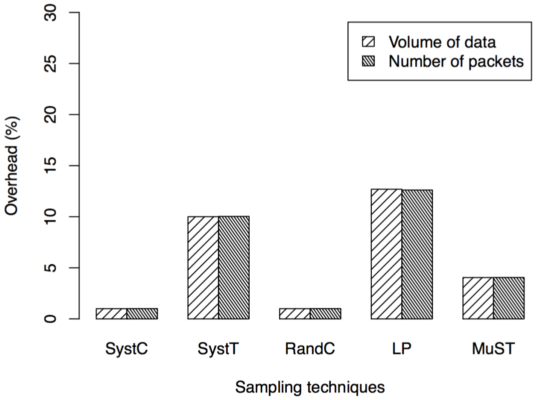


Fig. 3. OC-48: Data amount overhead

In order to compare the ability of sampling in representing the real traffic behavior, Figure 4 presents the accuracy results regarding identifying the total number of unidirectional flows. As expected, the techniques that sample larger volumes of data, identify a larger percentage of flows. However, when comparing count-based and time-based sampling techniques involving similar data volumes, i.e., SystC 1/32 with MuST and SystC 1/16 with SystT, time-based approaches reveal to be more effective. As example, SystC 1/32 detects less 9% of flows when compared to MuST, and SysC 1/16 leads to a decrease of 4% in flows identified when compared to SystT.

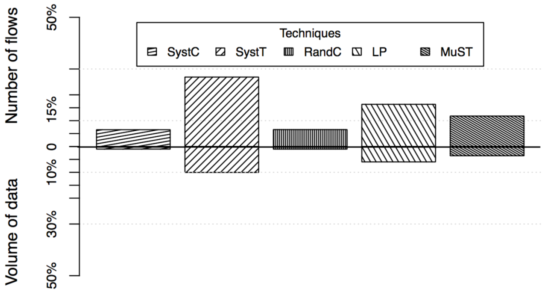


Fig. 4. OC-192: Flow identification per volume of data

The following analysis complements the accuracy in the number of flows identified, highlighting the flows duration and the impact the sampling techniques have on resource requirements. Figure 5 presents the box plot with the descriptive statistics of the number of active flows per time unit (sec). As shown, time-based techniques maintain a more accurate view of existing flows, although the large number of unsampled flows still affects significantly the instantaneous flow detection.

Complementing the flow identification comparison, the sampling accuracy analysis was extended to the context of traffic classification (both at transport and application level). As presented in Figure 6, all techniques overestimate TCP share, with count-based techniques exhibiting the highest *MSE - Mean Squared Error*. The classification at application level presents less variability in the results. As shown in Figure 6,

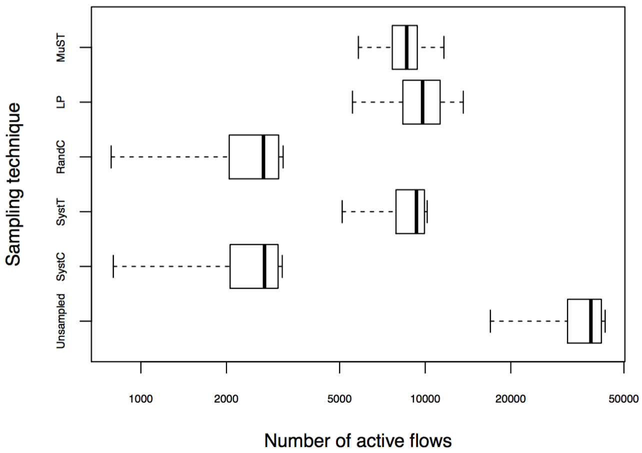
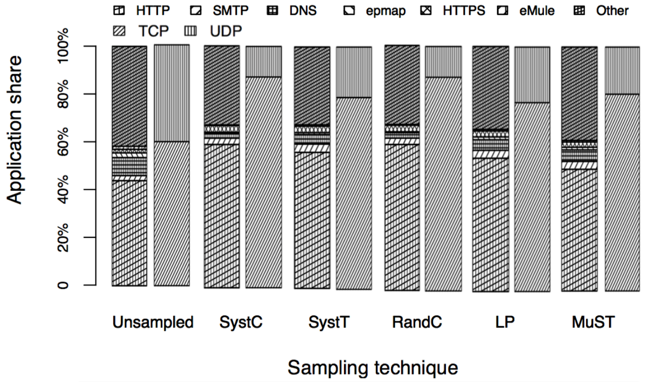


Fig. 5. OC-48: Statistics on the number of active flows



	SystC	SystT	RandC	LP	MuST
Transport level	0.68	0.23	0.68	0.18	0.29
Application level	0.32	0.15	0.31	0.10	0.07

Fig. 6. Application and Transport protocol share

time-based techniques lead to a more realistic distribution of the application share, with MuST providing a slightly more accurate result.

Taking an experimental distributed monitoring system at the Portuguese National Institute with nine Raspberry PI running the framework code, the analysis of sampling technique was extended by assessing their computational weight in terms of CPU load and memory consumption per MByte collected and stored for diverse workloads. As shown in Table II, in average, SysT and MuST achieve a better relationship for both resources, ratifying their advantage observed in flow analysis.

TABLE II
COMPUTATIONAL RESOURCE PER MBYTE

Ratio	SystC	SystT	RandC	LP	MuST
%CPU/MByte	1.40	0.20	1.73	1.71	0.45
%Memory/MByte	1.63	0.16	1.73	0.32	0.72

VI. CONCLUSIONS

Aiming to provide a common ground to support a flexible usage and test of current and forthcoming sampling techniques, this work proposed a modular framework based on a taxonomy of sampling techniques, allowing the study and deployment of versatile sampling-based measurement systems. Within a global measurement architecture, the developed framework is supporting research work related to the suitability of different sampling techniques and their computational weight when applied to various network activities. The results have showed that, although scarcely available in real measurement scenarios, some time-based and adaptive techniques provide relevant advances regarding flow identification and computational requirements (CPU load and memory usage) of classic approaches. The presented results exemplified the framework versatility and potential in fostering the deployment and tuning of new sampling techniques, revealing that a modular and configurable approach to sampling is a step forward for improving sampling scope and efficiency.

ACKNOWLEDGEMENTS

This work has been supported by FCT - *Fundação para a Ciência e Tecnologia* in the scope of the project: UID/CEC/00319/2013.

REFERENCES

[1] T. Zseby, M. Molina, and N. Duffield, “Sampling and Filtering Techniques for IP Packet Selection,” RFC 5475, IETF, Mar. 2009.

[2] Z.-G. Hu, D.-L. Zhang, C.-P. Hou, and J.-S. Zhang, “Adaptive sampling algorithm of network round-trip time,” in *Journal of Computer Applications*, J. Yingyong, Ed., vol. 30, no. 2, feb 2010, pp. 319–322.

[3] N. Duffield and M. Grossglauser, “Trajectory sampling for direct traffic observation,” *Networking, IEEE/ACM Transactions on*, vol. 9, no. 3, pp. 280–292, Jun 2001.

[4] D. Tammaro, S. Valenti, D. Rossi, and A. Pescapè, “Exploiting packet-sampling measurements for traffic characterization and classification,” *Int. Journal of Network Management*, vol. 22, no. 6, pp. 451–476, 2012.

[5] L. Yang and G. Michailidis, “Sampled based estimation of network traffic flow characteristics,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications.*, May 2007, pp. 1775–1783.

[6] M. Lee, N. Duffield, and R. Kompella, “Two samples are enough: Opportunistic flow-level latency estimation using netflow,” in *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1–9.

[7] S. Kandula and R. Mahajan, “Sampling biases in network path measurements and what to do about it,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ser. IMC ’09. New York, NY, USA: ACM, 2009, pp. 156–169.

[8] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, “Network anomaly detection and classification via opportunistic sampling,” *Network, IEEE*, vol. 23, no. 1, pp. 6–12, January 2009.

[9] T. Zseby, “Deployment of sampling methods for sla validation with non-intrusive measurements,” in *Proceedings of Passive and Active Measurements Conference*, Fort Collins, 2002.

[10] A. Dogman, R. Saatchi, and S. Al-Khayatt, “An adaptive statistical sampling technique for computer network traffic,” in *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, July 2010, pp. 479–483.

[11] Y. Gu, L. Breslau, N. Duffield, and S. Sen, “On passive one-way loss measurements using sampled flow statistics,” in *INFOCOM 2009, IEEE*, April 2009, pp. 2946–2950.

[12] J. M. C. Silva, P. Carvalho, and S. Rito Lima, “Enhancing traffic sampling scope and efficiency,” in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 5–6.

[13] E. Hernandez, M. Chidester, and A. George, “Adaptive sampling for network management,” *Journal of Network and Systems Management*, vol. 9, no. 4, pp. 409–434, 2001.

[14] J. M. C. Silva, P. Carvalho, and S. R. Lima, “A multiadaptive sampling technique for cost-effective network measurements,” *Computer Networks*, vol. 57, no. 17, pp. 3357 – 3369, 2013.

[15] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Solé-Pareta, “Analysis of the impact of sampling on netflow traffic classification,” *Computer Networks*, vol. 55, no. 5, pp. 1083 – 1099, 2011.