


A Modular Traffic Sampling Architecture: Bringing Versatility and Efficiency to Massive Traffic Analysis

João Marco C. Silva¹ · Paulo Carvalho¹  · Solange Rito Lima¹

Received: 2 May 2016 / Revised: 3 January 2017 / Accepted: 17 January 2017 /
Published online: 3 February 2017
© Springer Science+Business Media New York 2017

Abstract The massive traffic volumes and heterogeneity of services in today's networks urge for flexible, yet simple measurement solutions to assist network management tasks, without impairing network performance. To turn treatable tasks requiring traffic analysis, sampling the traffic has become mandatory, triggering substantial research in the area. Despite that, there is still a lack of an encompassing solution able to support the flexible deployment of sampling techniques in production networks, adequate to diverse traffic scenarios and measurement activities. In this context, this article proposes a modular traffic sampling architecture able to foster the flexible design and deployment of efficient measurement strategies. The architecture is composed of three layers—management plane, control plane and data plane—covering key components to achieve versatile and lightweight measurements in diverse traffic scenarios and measurement activities. Each component of the architecture is described considering the different strategies, technologies and protocols that compose the several stages of a measurement process. Following the proposed architecture, a sampling framework prototype has been developed, providing a fair environment to assess and compare sampling techniques under distinct measurement scenarios, evaluating their performance in balancing computational burden and accuracy. The results have demonstrated the relevance and applicability of the proposed architecture, revealing that a modular and configurable approach to sampling is a step forward for improving sampling scope and efficiency.

✉ Paulo Carvalho
pmc@di.uminho.pt

João Marco C. Silva
joaomarc@di.uminho.pt

Solange Rito Lima
solange@di.uminho.pt

¹ Departamento de Informática, Centro Algoritmi, Universidade do Minho, Campus de Gualtar, 4710-057 Braga, Portugal

Keywords Traffic sampling · Sampling techniques · Traffic measurement architecture · Traffic sampling taxonomy · Traffic monitoring and analysis

1 Introduction

In today's communication infrastructures, the massive volume of data, specially in high-capacity network links, is a major problem to overcome. This raises the necessity of reducing the volume of data involved in passive network measurements and providing sufficiently detailed information for various monitoring activities, with different accuracy requirements according to the service types or monitoring objectives to fulfill. In this context, traffic sampling is seen as the main strategy to keep network measurement data into manageable sizes, as it consists of selecting a subset of packets that will allow to estimate parameters about all traffic, with compatible degrees of accuracy, avoiding processing it completely.

Most of current measurement points (MPs), whether running in a dedicated device or embedded in switches or routers, provide tools able to perform traffic sampling following standard schemes defined by IETF (i.e., RFC5475 [1]). However, 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. Even tools following IETF recommendations, e.g., Cisco Sampled NetFlow and sFlow, usually do not implement all the features stated in RFC 5475. For instance, they do not provide crucial sampling strategies, such as time-based, relevant for activities related to anomaly detection [2]. Similar limitations are also present in different measurement tools and network vendors, for instance tcpdump, Alcatel cFlow, Juniper J-flow and Endace NICs. In fact, the lack of an encompassing study regarding the suitability of sampling techniques for multiple measurement tasks is limited by the specificities of new techniques and the complexity of deploying them in current commercial sampling tools, hampering thus a wide adoption of innovative and efficient sampling features.

Aiming at fostering the design and deployment of efficient sampling strategies for diverse measurement scenarios, this work is focused on understanding sampling techniques and measurement processes through their constituent parts, rather than closed units. This will empower the design of sampling systems with the ability to optimize their performance by exploiting the most suitable features for a specific measurement purpose or traffic type. In this context, this paper presents a three-layer modular sampling architecture based on a novel taxonomy of sampling techniques which allows the flexible deployment of different sampling techniques in a simple and modular way. Each layer is composed by independent components, and structured as a multilayer system in which a lower layer provides services to an upper layer, hiding details about its operation. Furthermore, the design of the architecture allows: (1) compatibility with currently deployed measurement systems; (2) flexibility to accommodate new measurement goals and traffic

characteristics; and (3) lightweight operation in order to minimize interference with the normal network tasks. This clearly constitutes an innovative approach toward efficient design and deployment of customized sampling.

This paper also discusses the strategies, technologies and protocols involved in the devised layers—Management, Control and Data—and provides a proof-of-concept exploring the flexibility of the proposed architecture in deploying and evaluating the tradeoff between overhead and accuracy of representative sampling techniques when facing different measurement requirements. For this purpose, real traffic traces captured in high-capacity network links of major service providers are used.

In summary, as main contribution, this article provides: (1) a clear identification of sampling components articulating them into a multilayer and modular architecture; and (2) a complete, flexible and easily configurable framework able to support seamlessly both conventional and innovative sampling-based measurements adequate to diverse network scenarios and monitoring activities.

This paper is organized as follows: the related work is reviewed and discussed in Sect. 2, the proposed sampling architecture, corresponding layers and components are described in Sect. 3; the proof-of-concept is provided in Sect. 4, the obtained results are discussed in Sect. 5, and the conclusions and future work are presented in Sect. 6.

2 Related Work

The usage of packet sampling aiming at fostering network measurements is not a recent research subject. The initial efforts addressing sampling techniques for statistical analysis of computer networks were mainly focused on QoS of communication systems, traffic accounting and characterization [3–5]. These early research works have produced methods to categorize sampling techniques [6], which have evolved to a framework standardized as a Request for Comments (RFC) [1] by the Packet SAMPLing (PSAMP) Working Group of the Internet Engineering Task Force (IETF) [7].

According to RFC5475 [1], sampling algorithms are classified in *content-independent* techniques and *content-dependent* techniques. The main difference between these classes is the necessity of accessing the packet content in order to make selection and capture decisions. Following this classification, the document also identifies the most deployed sampling schemes, namely: *systematic techniques* [7], which rely on deterministic functions based on the packet position in time or in space within incoming traffic; and *random techniques* [8], which resort to probabilistic functions in order to decide which packets will be selected to compose the sample. More complex techniques, such as *adaptive* [9] and *multiadaptive* [10], are not covered in this standard.

Simultaneously with the development of high-speed network infrastructures and the diversification of communication services, the usage of packet sampling has also increased significantly, leading to the support of manifold tasks related to network measurements. Examples of these tasks include: *network management* involving

short, medium and long term planning and management of network operation, maintenance and provisioning of network services [9, 11–13]; *traffic engineering* involving performance optimization, traffic characterization, traffic modeling and control [4, 7, 14–16]; *performance evaluation* of protocols and management tools, network reliability and fault tolerance [17–19]; *network security*, including detection of anomalies, intrusion, botnet and Distributed Denial of Service (DDoS) attacks [20–25]; *SLA compliance*, where auditing tools may resort to network sampling for measuring and report service levels [26, 27]; *QoS control*, aiming at measuring parameters such as delay, jitter and packet loss [28–31]. Most of the above cited techniques are conceived resorting to modifications and/or composition of basic approaches that compose classical sampling techniques.

Beyond the vast number of new techniques, the diversification of communication services and their underlying requirements have also fostered a large number of research works focused on assessing the most suitable sampling technique for different measurement tasks. Several of these studies are mainly devoted to analyze and enhance the trade-off between sampling accuracy and overhead of the traditional techniques defined in [1], which essentially intend to minimize information loss while reducing the volume of collected data [32].

From these works, it is clear that the selection of a strategy for packet sampling depends on the type of measurement task in which it will be applied. Although PSAMP proposes a high-level architecture description allowing a modular definition of sampling, it is mainly focused on packet selection and exporting, lacking of comprehensiveness and objectivity regarding the components and solutions for relating measurement tasks with sampling techniques. Furthermore, PSAMP-based implementations (e.g., Sampled NetFlow, sFlow) are not modular and restrictive in the techniques implemented. In this way, devising an encompassing sampling-based measurement system able to accommodate different sampling strategies aiming at providing high-accuracy measurements while maintaining the computational overhead under control in distinct measurement scenarios is not only an open issue, but also highly desirable.

3 A Sampling-Based Measurement Architecture

The design of an encompassing and flexible packet sampling architecture able to foster the deployment of versatile and lightweight measurement strategies in diverse traffic scenarios and measurement activities must satisfy the following design goals:

- compatibility with current protocols and measurement tools in order to support its deployment in current measurement systems;
- specification and deployment sustained by open and standard protocols;
- flexibility to adopt new protocols and technologies related to traffic sampling;
- versatility and modularity to deploy current and new sampling techniques;
- capability to support mechanisms for balancing measurement accuracy and computational weight in order to foster the design of efficient sampling strategies.

The main components involved in the proposed sampling-based measurement architecture are arranged in three planes—*management, control and data plane*—as illustrated in Fig. 1. These components are further detailed in the next sections, considering the different strategies, technologies and protocols involved in the several stages of a measurement process.

3.1 Management Plane

The main activities assigned to the management plane are: (1) mapping the measurement needs related to a specific network task into the more suitable sampling technique and its operational parameters; (2) selecting and communicate with the MPs which will perform packet sampling in order to set them up; (3) processing the measurement results and provide a visualization component, when applicable, based on reports produced by the control plane. These functions may be deployed directly

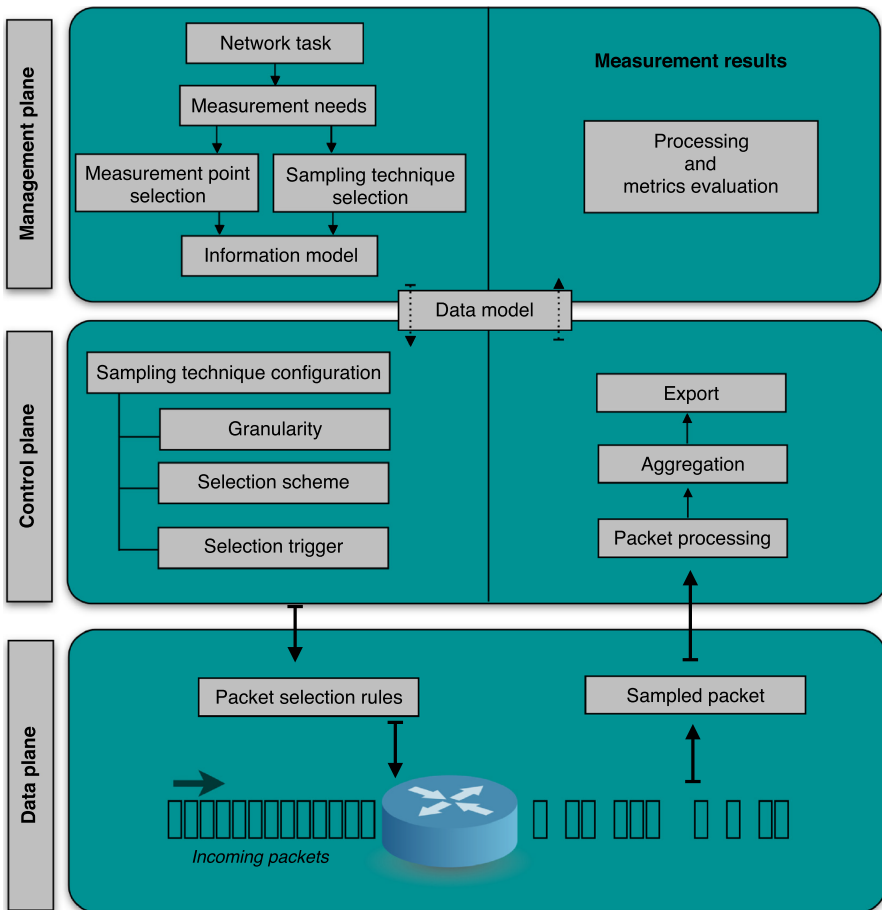


Fig. 1 Architecture description

into the MP, sharing the same device and resources, or in an external entity, responsible for coordinating one or more MPs according to measurement needs and constraints.

3.1.1 From Network Task to Measurement Needs

Measurement needs are closely related to the network task to fulfil. This relation usually guides the sampling process, defining aspects such as: (1) which portion of the packet should be captured and exported; (2) if the sampling should be performed considering all incoming traffic or only specific flows; (3) the temporal and spatial distribution of the packet collection; or (4) the expected accuracy on metrics estimation.

Some network tasks, such as traffic accounting, only require few information from the packet header, usually the key flow, the number of packets and number of bytes traversing the MP during a certain time interval. These requirements tend to be less impactful in terms of storage and bandwidth, as traffic may be aggregated into flows efficiently before exporting. However, considering network tasks which resort to Deep Packet Inspection (DPI), such as traffic classification and data analysis for security issues, the MP must inspect and collect the packet payload in addition to the header. This also involves exporting individual packets instead of aggregated summaries, which may lead to a large volume of data related to measurements transmission. The measurement needs may also vary depending on the expected accuracy in metric estimations, for instance, the accuracy in estimating the traffic workload is affected by the sampling frequency [33].

The relation between the network task and its measurement needs should drive directly the decision of the sampling technique and the MPs to be used. Despite this mapping being out of scope in this work, the next section and Sect. 5 present important aspects to be considered and highlights possible strategies.

3.1.2 Sampling Technique and MP Selection

Currently, due to the small number of sampling techniques available in measurement tools, the selection of the technique and its operational parameters are always a decision of network managers. Although there is not an encompassing study addressing which sampling technique yields better results for each network task, by reviewing the related literature it is possible to identify that the results achieved by different works are clearly heterogeneous and, sometimes, conflicting [24, 34–36].

Other possible conflicting aspect is related to sampling measurements from different network tasks performed in the same MP during a time interval. This aspect may be addressed resorting to a priority system, in which the decision from the task with highest priority level prevails in the MP, or configuring the most demanding technique (i.e., which captures the largest amount of data) in order to accommodate the largest possible number of network tasks efficiently.

Regarding the selection of the MPs that will participate in the sampling process, this may involve a single point, two points (e.g., end-to-end delay) or a distributed multipoint strategy. This decision should take into account the position of the device

in the network, the computational resources available and the sampling techniques able to be deployed (for legacy tools). There are many studies addressing the selection of the better network point to perform each type of measurements [37, 38], as an appropriate strategy of MPs selection leads to more efficient use of resources and may reduce the events of conflicting configuration. The example in Fig. 2 illustrates a possible relation between the network task and the MPs selection, namely: (1) traffic accounting in this topology could involve only MP-A (sampling the external link of the border router); (2) QoS multipoint metrics estimation, such as one-way-delay and jitter, could involve the edge MPs, for instance MP-C and MP-E; (3) security-oriented measurements would probably require data sampled by all MPs.

3.1.3 Information Model

As described in RFC3444 [39], the main purpose of an information model is to define managed objects at a conceptual level, independently of specific implementation or protocol used to transport data. In this way, the information model is defined at management plane as a standardized way for encoding information related to the sampling process (e.g., technique selected, sampling parameters and packet fields to be collected), exporting and storage of sampled data.

As result of the efforts toward the definition of an open protocol for flow exporting [40], the IETF IPFIX—*Internet Protocol Flow Information Export* working group has also defined an information model (RFC7012 [41]) that was further extended to satisfy PSAMP requirements through RFC5477 [42]. Proposing an extended model was necessary due to existing properties required in packet sampling reports that cannot be modeled using the basic IPFIX information model.

The information model is composed by unique identifiers related to each *information element*, that consists in an encoding-independent description of an

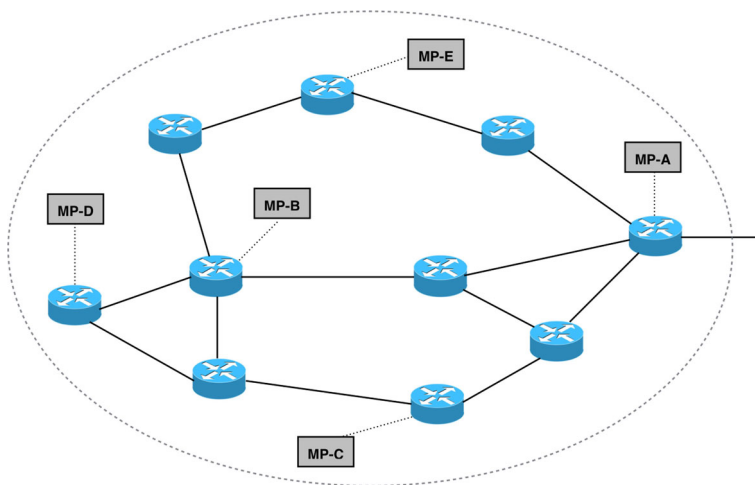


Fig. 2 Measurement point selection

attribute that may appear in a measurement record. The information elements also have an associated type, that indicates constraints on what it may contain as well as the valid encoding mechanisms [41]. The information element assignments are controlled by IANA—*Internet Assigned Numbers Authority* and are exemplified in Table 1. A full list of elements currently assigned can be consulted in [43].

3.1.4 Data Model

Data models define managed objects at a lower level of abstraction, including implementation aspects and protocol specifications, such as the rules that explain how to map managed objects onto lower-level protocol constructs [39]. The definition of a common data model is required to allow managing the entities in the sampling process.

In addition to the specific information model for packet sampling, the IETF IPFIX working group has also proposed a standard that defines managed objects for monitoring devices performing packet selection by sampling (RFC6727 [44]). The document is in accordance with the Internet-Standard Management Framework [45], in which managed objects are stored in a Management Information Base (MIB) and generally accessed through SNMP. The syntax used to define objects in the MIB is called the Structure of Management Information (SMI). Currently, the working group is defining a standard with a method for exporting SNMP MIB variables using IPFIX messages [46].

There are other management models currently standardized as data models, for instance the Policy Information Base (PIB) [47] and the Common Information Model (CIM) Schemas [48]. However, the use of open standard models designed to support packet sampling leads to a straightforward integration with current tools able to control MPs and process the resulting sampled data reports.

3.1.5 Processing and Metrics Evaluation

As discussed in Sect. 3.2, the packet sampling process yields different types of reports that must be processed for further estimation of underlying metrics regarding the network task. This is usually performed by a *collector*, that may be an intermediate entity responsible for verifying and distributing the reports to interested entities or an application able to provide summarized measurement results.

Table 1 Example of IPFIX information elements

Element ID	Name	Data type	Data type semantics
1	octetDeltaCount	unsigned64	deltaCounter
2	packetDeltaCount	unsigned64	deltaCounter
304	selectorAlgorithm	unsigned16	identifier
309	samplingSize	unsigned32	quantity

Following IETF definitions [1], the collector receives a *report stream* by one or more MPs. The report stream comprises two types of information: (1) *packet reports*—a configurable subset of packet’s data regarding the measurement needs (e.g., packet content); and (2) *report interpretation*—a subsidiary information used for interpretation of the packet reports (e.g., templates describing their structure and types). An example of both report types is presented in Sect. 3.2.5 (Fig. 4).

Supported by the definition of a consistent information model and data model, the entities involved in sampling can process the reports and access the required traffic information in order to estimate measurement metrics.

3.2 Control Plane

The control plane is the main component of the proposed architecture. It is responsible for: (1) selecting and arranging the constituent parts of the sampling technique to be deployed, defined by the management plane, and setting its operational parameters; (2) receiving raw packets collected by the data plane and extract required information regarding the measurement needs; (3) aggregating data in order to reduce the storage and transmission impact; (4) composing the appropriate reports to be sent to the management plane.

3.2.1 Constituent Parts of Sampling Techniques

Describing current sampling techniques through its constituent parts, rather than a closed unit, is a key aspect to achieve flexible sampling-based measurements. Although classic sampling techniques were previously classified through RFC5475 [1], this document does not cover recent advances in packet sampling (e.g., adaptive techniques). In this way, after a detailed analysis of existing sampling proposals, the present work presents an encompassing sampling taxonomy, identifying the constituent parts of these proposals. Table 2 details the proposed taxonomy, following the preliminary classification included in [49].

In the classification criterion, a set of features related to sampling *granularity*, *selection scheme* and *selection trigger* are identified as the main components distinguishing current proposals. Then, each component is further divided into a set of approaches. Following the taxonomy structure it is possible to drive a modular deployment of sampling techniques through the combination of proper approaches. Beyond the comprehensiveness in classifying current sampling techniques, this model allows the design of sampling strategies able to exploit specific features that lead to better performance for each measurement purpose and traffic scenario.

3.2.2 Sampling Technique Configuration

Following the guidelines presented in Sect. 3.1.3, a MP receives the necessary information in order to select and configure the sampling technique that will supply the management plane according to the network task requirements. Table 3 presents examples of sampling techniques, their underlying operational parameters and the respective element identification according to IANA scope assignments. As

Table 2 Taxonomy of sampling techniques

Granularity		
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.		
<i>Flow-level</i>	<i>Packet-level</i>	
The traffic capture policy is applied to packets belonging to a flow or a set of flows of interest. This involves classifying packets into flows before or during the sampling process through the identification of a flow key, usually based on five fields (5-tuple) of the packet header.	In a first instance, packets are collected indistinctly for subsequent filtering or aggregation. This turns packet-level sampling into a flexible and appropriate solution to be used in general purpose measurement tasks and aggregated estimations, in presence of diverse traffic types.	
Selection trigger		
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.		
<i>Count-based</i>	<i>Time-based</i>	<i>Event-based</i>
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. This strategy is used in Sample NetFlow and sFlow.	The beginning and the end of a sample is determined based on packet arrival time. When a new sample is triggered, the MP waits for the first bit of the incoming packet to start the collection. When sampling end is triggered, the MP waits for the last bit of the current packet and then stops the selection process.	The decision on when a sample starts and ends takes into account some particular event observed in the traffic being monitored. This event might be some value in the packet contents e.g. packet header and payload, the treatment of the packet at the measurement point or a more complex observation.
Selection scheme		
This component identifies the selection function that determines which packets will be selected and collected.		
<i>Systematic</i>	<i>Random</i>	<i>Adaptive</i>
The process of packet selection is ruled by a deterministic function which imposes a fixed sampling frequency, independently of the packet content or treatment. In this scheme only equally spaced traffic is collected, i.e., the sampling trigger is periodic.	The sampling frequency is ruled by a random process using for instance n-out-of-N and probabilistic sampling [1]. The probabilistic function can be uniform or non-uniform.	The selection process is able to change the packet selection criterion during the course of measurements. Adaptive techniques usually resort to linear prediction, fuzzy logic or other strategies that consider traffic behavior, packet content or network status to rule sampling patterns.

illustrated, the sampling technique selected is identified by the information element *selectorAlgorithm*, identified in the information model with the value 304. Each sampling technique has a set of well-know parameters (also defined in the information model), that must be passed along with the technique identifier. Handling this information, the control plane starts an instance of the technique with the respective operational parameters.

The internal process for deploying a sampling technique follows the structure of the proposed taxonomy. By selecting the appropriate approach from each sampling component, the control plane arranges the respective blocks, configuring thus the selected technique. This process requires well defined communication interfaces among the sampling approaches. This is achieved designing the *sampling framework* as a multilayer system in which a lower layer provides services to an upper layer, hiding details about its operation. Figure 3 illustrates the conceptual

Table 3 Sampling technique identification example

ID	Sampling technique [selectorAlgorithm (304)]	Parameters (ID)
1	Systematic count-based [1]	samplingPacketInterval (305) samplingPacketSpace (306)
2	Systematic time-based [1]	samplingTimeInterval (307) samplingTimeSpace (308)
3	Random n-out-of-N [1]	sampleSize (309) samplePopulation (310)
10 ^a	Multiadaptive [10]	samplingTimeInterval (307) samplingTimeSpace (308)
11 ^a	Flow-level adaptive linear prediction [9]	flowId (148) samplingTimeInterval (307) samplingTimeSpace (308)

^aAs these techniques are not yet assigned by IANA, the examples use currently unassigned values, avoiding conflicts with deployed tools

design of this framework. Packet capturing is performed resorting to an interface with the data plane, being its operational details covered in Sect. 3.3.

3.2.3 Packet Processing

As discussed in Sect. 3.3, to reduce the computational burden in the data plane, the sampled packets are received by the control plane in raw format, for verification and processing. The verification process allows to identify errors in the packet that may have occurred during the handover from the capture interface to the upper plane of the network stack. Error detection can be performed resorting to any available method, such as checksum or cyclic redundancy checking. As error correction is a computationally onerous process, if an error is found then the packet is discarded.

Considering that packets are received in raw format, it is also necessary for mapping the packet fields to the suitable data model in use by the measurement system. The mapping process is supported by the information model in order to unify the elements representation, allowing the correct interpretation and manipulation of the packet fields.

At the processing stage, all irrelevant fields to measurements are discarded, reducing the amount of data received and processed by the upper modules, and consequently the number of computation cycles, bandwidth and memory to process the sampled traffic.

3.2.4 Aggregation

Even with the reduction in the volume of data promoted by sampling, some scenarios can still produce massive data amounts, requiring significant storage resources and bandwidth to be distributed. A solution to this issue is to summarize the collected data employing a combination of sampling and aggregation. The strategy mostly used to summarize measurement data is to aggregate sampled packets into flows according to some explicit or derived property, and computing

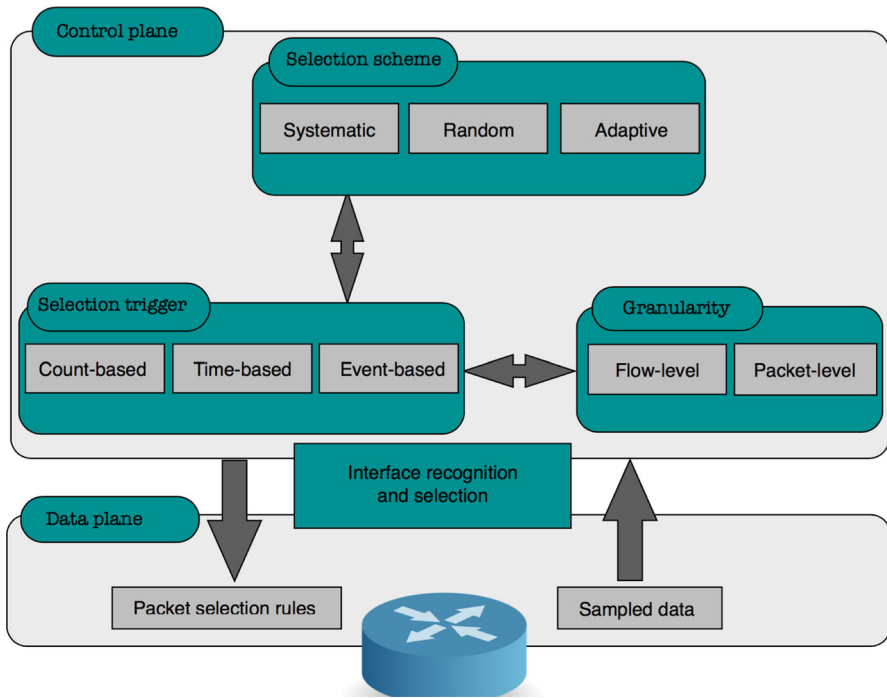


Fig. 3 Sampling framework—conceptual design

aggregate byte and/or packet counts within each flow over successive time windows [8]. This method usually provides high accuracy when estimating metrics that only require information from the packet header, such as for traffic accounting, flows distribution and anomaly detection. However, there is an increasing need for analyzing packet payloads (DPI) in network tasks such as traffic classification and security analysis. In fact, IANA has been registering information elements describing objects related to the application layer. In these scenarios, as packets are typically stored as individual entities, and thus hampering data summarization, the aggregation module does not act on packets, maintaining them in memory for future exporting.

3.2.5 Exporting

The exporting process consists of three main elements: (1) the trigger for dispatching data currently stored in the MP memory; (2) the message structure and types used to transmit the data; (3) the transport protocol used in the data report transmission.

Exporting trigger Sampled traffic aggregated into flows is frequently maintained in memory until a specific flow is considered to have terminated, after which, the information regarding this specific flow is exported. The flow expiration may follow different methods: a natural termination of a TCP flow when a packet with a FIN or

RST flag set is captured; the flow has been active for a specific period of time (usually within the range between 120 seconds to 30 min); or no packet belonging to a flow is captured during a specified period of time (usually between 15 s and 5 min) [50]. In addition to predefined timeouts, resource constraints may require strategies in which the timeout is dynamically adjusted in run-time.

Considering measurements involving the exporting of the packet payload, a MP may also use predefined timeouts or a dynamic strategy based on the volume of packets stored in memory. Moreover, this strategy is also suitable to trigger the exporting of both aggregate and single packet entries in scenarios of full memory or in response to unexpected situations.

Message format The distribution of sampled data must be supported by well defined message formats in order to ensure that the applications involved in measurements can interpret correctly the information. There are well defined protocols and tools able to handle with this aspect, such as NetFlow, SiLK and IPFIX. Reports using XML specifications are also frequently used. In particular, the IPFIX specification has a specific version [51] designed to address the architectural differences between the original version [40], focused on gathering and exporting IP traffic flow information, and the PSAMP extension, focused on exporting information of individual packets. Sampling-based reports are therefore a special IPFIX record containing only a single packet.

The IPFIX message can be of two types: (1) *template record*, that contains the layout description for data report interpretation; and (2) *data report*, used for carrying exported data records [50]. For each collected packet, a data report must be created containing a header with a set of fields with fixed size (16 bytes), identifying the protocol version number, message length, export timestamp and the observation domain ID. After the header, one or more sets (i.e., one or more records), are defined having an ID and variable fields.

Figure 4 presents a simplified example of a template record message and its correspondent data report, identifying that the packet, with 64 bytes, was collected

(a) <i>Template</i>		(b) <i>Data</i>	
Version (2bytes)	Length (2bytes)	0x000a	42
Export time (4bytes)		2015-08-01 21:25:02	
Sequence number (4bytes)		0	
Observation domain ID (4bytes)		132435	
Set ID = 2 (<i>template</i>)	Length = 16	256	24
Template ID = 321	Number of fields = 4	321	4
selectorAlgorithm (ID = 304)		1	
samplingPacketInterval (ID = 305)		1	
samplingPacketSpace (ID = 306)		99	
octetDeltaCount (ID = 1)		8	

Fig. 4 IPFIX messages—template and data, **a** template record, **b** data report

through *systematic count-based* sampling technique capturing 1 packet from every 100. All elements in a report are specified by the information model (see Sect. 3.1.3) and each template has an unique ID, allowing that all entities involved in the measurements can interpret the data reports following this template. Usually the number of records in an IPFIX message is limited in order to avoid IP fragmentation [50].

Transport protocol The selection of a transport protocol to transmit the measurement reports should consider the collector/application or MP restrictions. The usual candidates are User Datagram Protocol (UDP), Transmission Control Protocol (TCP) and Stream Control Transmission Protocol (SCTP). Due to the easy implementation (even in hardware) and minimal overhead, UDP is the most implemented transport protocol for measurement data transmission. A discussion on advantages and drawbacks of each solution can be found in [50].

3.3 Data Plane

At data plane, following the sampling rules defined in the control plane, packets are collected from the network link for subsequent use.

In wired networks, where most traffic measurements are performed, the MP implements an interface (also called capture device) in which it is possible reading and collecting packets from the link being monitored. The capture interface can be positioned *in-line* and in *mirroring* mode. While in *in-line* mode, the MP is directly connected to the monitored link between two hosts, usually resorting to a network tap that duplicates all observed traffic through passive splitting (on optical fiber links) or regeneration (in electrical copper networks), in *mirroring* mode, the network device forwarding packets can mirror packets from one or more ports to another port, in which the MP device is attached.

In wireless networks, the MP may use any device with a compatible interface (usually these devices can only capture packets at a single frequency at a given time [50]), however some of them can switch rapidly through all radio channels (channel hopping) trying to improve traffic capturing, although there is no guarantee that all packets are considered [52]. In virtual networks, the nature of the devices is similar to wired networks, although in this case the capture interfaces are usually entirely deployed in software.

In addition to the device nature and location, the data plane also defines how the control plane interacts with the network interface in which the packet capture is performed. For this, MPs resort to libraries and Application Programming Interfaces (APIs) in order to implement the packet collection. The main solutions currently available are using *libpcap* [53] or *libtrace* [54] for Linux and BSD-based operating systems, and *WinPcap* for Windows.

4 Proof-of-Concept

Aiming at providing a proof-of-concept regarding the flexibility introduced by the sampling architecture, a functional framework able to deploy different sampling techniques following the taxonomy structure presented in Table 2 has been

implemented. This framework provides a fair environment in which different sampling techniques can be comparatively assessed in order to identify the most suitable for each measurement goal and traffic scenario. This is a fundamental aspect in order to support the design of efficient measurement strategies based on packet sampling.

The developed sampling framework¹ 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 [55]. In particular, the experimental tests reported in this work evaluate the impact of different sampling strategies regarding the volume of collected data and accuracy in estimating traffic workload and performing flow analysis in high-speed networks.

4.1 Traffic Scenarios and Sampling Techniques

The performance analysis carried out resorts to real and public traffic traces captured in high-speed network links, namely OC-48 [56] and OC-192 [57]. 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) [9]; and MuST—Multiadaptive (time-based) sampling [10]. The following comparative evaluation uses the frequency 1/100 for SystC and RandC techniques, as suggested in [33]. For SystT technique, the sampling frequency in use is 100/1000.

4.2 Comparative Parameters

4.2.1 Volume of Data

The main goal of using traffic sampling consists in to reduce the overhead associated with the amount of packets processed, which may impact on the overall performance of the MP, the bandwidth required to export measurement data as well as the storage and processing overhead [33]. In this way, the sampling techniques are compared regarding: (1) *Number of packets*—total number of packets captured during the sampling process for each sampling technique; (2) *Volume of data*—sum of all packets collected with each sampling technique. For this metric, it is used the total length field within IP header.

¹ The framework is available for download at <http://1drv.ms/1HgkCa> as a Raspbian image ready to be deployed.

4.2.2 Traffic Workload

The accuracy in estimating traffic workload is analyzed through the *mean throughput*, i.e., the total estimated load during the full sampling process, quantified by the Relative Mean Error (RME). Furthermore, the *mean packet size* and complementary descriptive statistics to measure the variability of packet time series are also analyzed.

Considering that in traffic sampling only a subset of total network packets is captured and considered for measurement purposes, estimating the traffic throughput must consider the unselected packets. The most common method to estimate the mean throughput from sampled data resorts to the statistical extrapolation based on the proportional number of unsampled packets, as detailed in [33], i.e., $\bar{X} = \frac{(\sum_{i=1}^n X_i) * S_p}{\Delta T}$. In this equation, \bar{X} is the estimated mean throughput; X_i is the size of the i th sampled packet; S_p is the statistical sampling proportion defined by $S_p = \frac{m}{n}$, with m as the total number of arriving packets and n the total number of sampled packets; and ΔT is the period of observation in seconds. In this work, the mean throughput is estimated taking ΔT equal to the total period of the sampling process.

4.2.3 Flow Analysis

For comparing the ability of distinct sampling techniques in assisting network flow analysis correctly, several flow parameters are considered, namely: (1) the amount of flows identified; (2) the percentage of heavy-hitter (HH) flows identified, where the notion of heavy hitter refers to 20% of the largest flows (in number of packets) [58]; (3) the utilization share at transport level; (4) the utilization share at application level; and, (5) the accuracy of load estimations for the identified flows.

Considering that when flow characterization is based on sampling only a subset of the packets is available, estimating the underlying metrics involves the usage of statistical estimators to overcome missing data. In particular, the load estimation of each flow is an additional challenge as it needs to be often inferred from a small number of collected packets. Following the discussion in [33], the specific estimators used for comparative purposes are the following:

- *Flow Mean Packet Size* (\bar{X}_f): $\bar{X}_f = \frac{\sum_{i=1}^{n_f} X_i}{n_f}$
- *Estimated Flow Size* (S_f): $S_f = N * \frac{n_f}{n_s}$
- *Estimated Flow Load* (L_f): $L_f = S_f * \bar{X}_f$

where X_i is the size of the i th sampled packet of flow f ; n_f is the number of sampled packets of flow f ; n_s is the total number of sampled packets; and N is the estimated total number of packets (n_s /sampling_frequency).

Regarding the estimated flow load, this work applies an innovative way to assess accuracy by resorting to a nonparametric method to estimate the density distribution of load estimation (i.e., KDE— Kernel Density Estimation method) and thereby

fostering the discussion on the estimation bias when applying each sampling technique. Each distribution corresponds to a nonparametric probability density function estimated using the Kernel method and a Gaussian smoothing scale. This method consists in drawing a continuous and smooth density distribution, weighted by the distance from a central value (the Kernel), where the population is inferred from a finite number of observations. In this context, as defined in [59], let (L_{f1}, \dots, L_{fn}) be the estimated load of all identified flows (n) for which the density p is under evaluation. The shape of this function using the kernel estimator is given by:

$$\hat{p}_{bw}(L_f) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{L_f - L_{fi}}{bw}\right), \quad (1)$$

where $K()$ is the kernel scaled by a Gaussian function, and bw is a smoothing parameter called bandwidth which defines the variance of the kernel in order to concentrate the density distribution within a specific interval. This interval is defined using the standard deviation of the smooth kernel when considering both unsampled traffic and traffic resulting from all sampling techniques.

When useful, the present study includes the mean absolute error (MAE) and the mean square error (MSE) of the estimated values, which are commonly used to evaluate the accuracy of estimators [33].

5 Evaluation Results

The results reported in this section evaluate comparatively the different sampling strategies regarding: (1) the volume of sampled data; (2) the traffic workload estimation accuracy; and (3) the ability to identify and classify network flows. The section ends highlighting the major findings.

5.1 Volume of Data

Regarding the volume of data collected and stored along the sampling process, Fig. 5 presents the percentage (relative to the unsampled trace) of the number of packets and volume of data for OC48 (Fig. 5a) and OC192 (Fig. 5b) traces. As shown, the count-based techniques use less resources when comparing to all time-based techniques under analysis. It is important to observe that count-based techniques allow a previous definition of the proportion of the total traffic that will be collected (in number of packets). This makes these techniques more suitable for MPs with storage limitations or for reducing the impact of measurement data traversing the network during the sampling exporting process.

Taking into account the sampled data unpredictability in time-based approaches, MuST technique has demonstrated to be more efficient in this selection trigger group. The results also show a close relation between the two parameters (*i.e.*, number of packets and volume of data) for all traffic scenarios and sampling

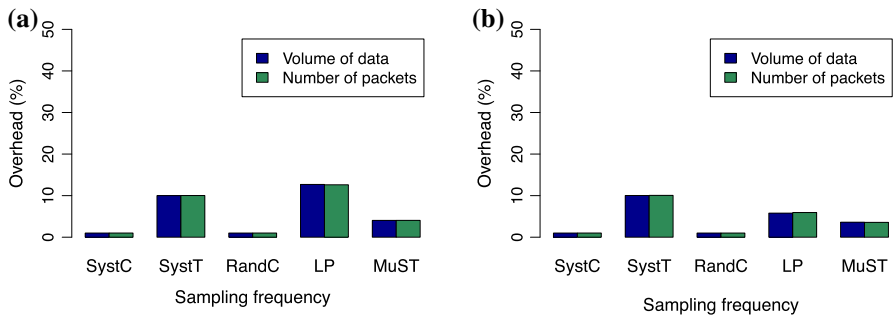


Fig. 5 Comparative data volume—traces from CAIDA, **a** OC-48, **b** OC-192

techniques. For instance, in trace OC48, SystC has captured 1% of the original packets corresponding to 1.2% of the total data amount.

5.2 Traffic Workload

Considering the comparative analysis of traffic workload, Table 4 presents the mean throughput estimated after each sampling process. The mean throughput analysis of different sampling techniques is particularly useful to guide measurement strategies for activities such as traffic engineering, accounting and SLA compliance, as measurements over large time intervals are considered. In general, for all traffic scenarios, the RME is low (less than 5%). The exception is the LP technique, with a relative error above 10% for both scenarios. In this regard, the comparison between the remaining time-based and count-based techniques also does not show significant variation, indicating that the lower performance of LP is related to its adaptive selection scheme.

Analyzing the packet size distribution, Table 4 shows that all techniques achieve accurate estimations of mean packet size. The ratio between peak and average packet size, a descriptive statistic to measure the variability of packet time series,

Table 4 Overall traffic behavior—all sampling techniques

Parameter/scenario	Total	SystC	SystT	RandC	LP	MuST
OC48						
Mean throughput (Mbps)	975.97	979.38	978.43	979.92	1085.12	985.29
RME	–	0.0035	0.0025	0.004	0.1118	0.0095
Mean packet size (Bytes)	565.44	567.31	566.84	567.94	561.71	566.18
Peak-to-average	3.67	3.58	3.66	3.65	3.69	3.66
OC192						
Mean throughput (Mbps)	1534.78	1535.57	1534.52	1537.07	1308.13	1515.29
RME	–	0.0005	0.0001	0.0014	0.1476	0.0126
Mean packet size (Bytes)	626.50	627.14	626.91	627.42	623.98	617.54
Peak-to-average	1.89	1.89	1.89	1.90	1.85	1.90

for identifying burstiness, also ratifies the estimation accuracy of all techniques. OC192 traffic exhibits the lower variability along the measurement process, a feature correctly identified by the sampling techniques.

5.3 Flow Analysis

In order to compare the ability of sampling in capturing the real traffic behavior, Fig. 6 presents the accuracy results regarding the identification of 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 an example, SystC 1/32 detects less 9% of flows when compared to MuST, and SystC 1/16 leads to a decrease of 4% in flows identification when compared to SystT.

Detailing these results, Table 5 illustrates the accuracy of the sampling techniques in identifying the most representative flows (heavy hitters). As shown, time-based techniques achieve better results in identifying the most representative flows, regarding volume of data. This is explained by the intrinsic nature of time-based techniques in capturing successive packets during a sampling interval. As heavy hitters flows tend to comprise a larger number of packets in that interval, the probability of being identified increases. This behavior is even more evident when considering the top 5% heavy flows.

Attending to the formulation in Sect. 4.2.3, the results in Fig. 7 show the distribution of the estimated flow load L_f (in logarithmic scale) when applying the different sampling techniques. The resulting graphics demonstrate the ability to represent the load distribution of all flows identified in the traffic trace, instead of only the more significant ones. This analysis plays a key role for traffic characterization and resource management activities.

The results show that time-based techniques achieve a distribution closer to the real flow behavior (unsampled case in Fig. 7a) when compared with the count-based

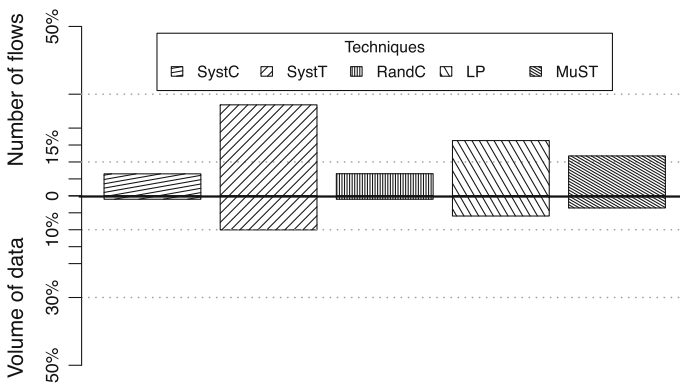


Fig. 6 OC-192: flow identification per volume of data

Table 5 Accuracy in identifying representative flows

	SystC (%)	SystT (%)	RandC (%)	LP (%)	MuST (%)
OC-48	8.34	27.09	8.39	31.39	28.19
OC-192	8.95	29.12	8.99	18.98	27.45

approaches (Fig. 7b, d), due to their more accurate load estimations of individual flows. This is observed through the better adjustment on the x-axis, meaning that the load estimations are closer to the real values. This suggests that a positive aspect (sparse packet selection) in flow identification becomes a drawback of the count-based techniques in flow dimensioning, since the current heuristics for flow load estimation is linear extrapolation proportional to the sampling frequency. This may

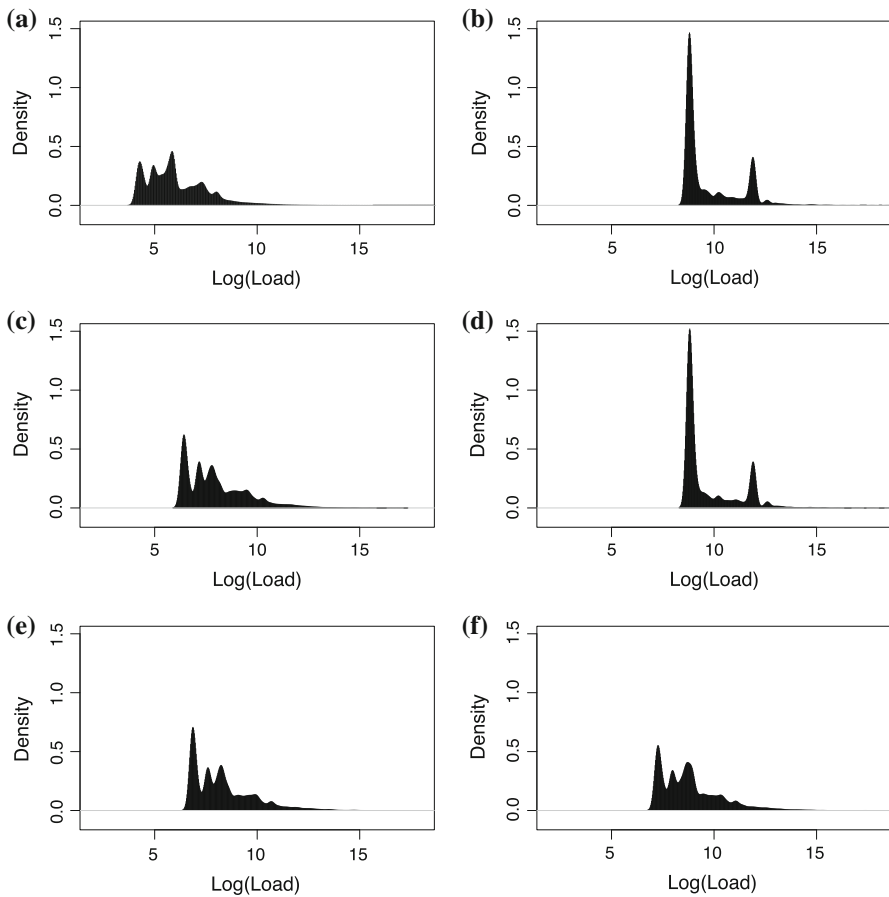


Fig. 7 OC-48: density of flow load estimation, **a** unsampled , **b** SystC , **c** SystT , **d** RandC , **e** LP , **f** MuST

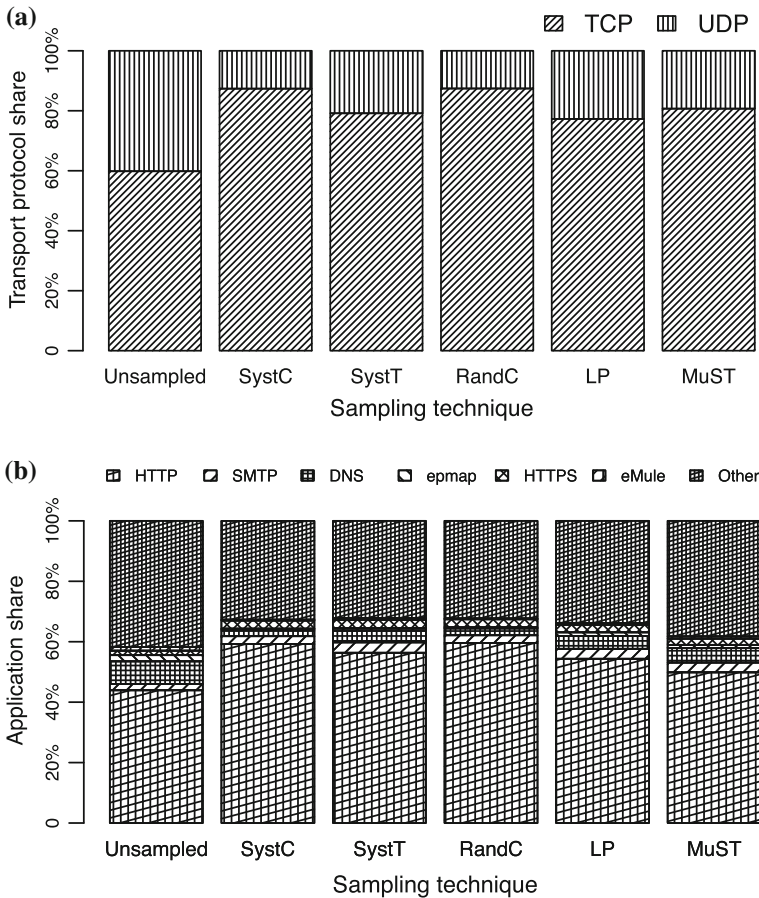


Fig. 8 OC-192: accuracy in flow classification, **a** analysis at transport level , **b** analysis at application level

interfere with network tasks in which the classification of small flows are of particular interest, such as intrusion detection and DDoS attacks.

Conversely, once time-based techniques select successive packets, the bursty behavior of flows tends to be better identified and dimensioned (when occurs within the sample size interval), resulting in more accurate flow load distributions, as presented in Fig. 7c, e, f.

To complement this analysis, the sampling accuracy analysis was extended to the context of traffic classification (both at transport and application level). As presented in Fig. 8a, all techniques provide fairly accurate estimations of the most significant transport protocols in use (TCP, UDP), with LP exhibiting lower Mean Squared Error (MSE), as shown in Table 6. The classification at application level²

² Note that the evaluation of flow classification methodologies and tools is beyond the scope of this work, which resorts to a port-based classification technique for distinguishing flows.

Table 6 MSE OC-192: traffic classification

	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

presents more variability in the results. As shown in Fig. 8b and quantified in Table 6, time-based techniques lead to a more realistic distribution of the application share, with LP and MuST providing a slightly more accurate result. Globally, the results evince that an adequate yet small fraction of network traffic is able to provide a useful panoramic view of the protocolar mix of network flows.

5.4 Evaluation Remarks

Although the present evaluation study considers a limited set of possible measurement activities, the obtained results bring a valuable comparative insight among existing sampling techniques, providing a better understanding of their suitability and overhead in real scenarios. The results showed that despite the extensive deployment of systematic and random count-based techniques in current measurement tools, the adaptive and systematic time-based techniques can outperform them in important aspects, such as the accurate flow identification and dimensioning.

The lack of a conclusive best sampling technique (in terms of overall performance), even considering a small set of measurement goals, ratifies the central purpose of this work. The performed analysis also demonstrated that, for some measurement goals, it is possible to reduce the amount of data involved in the network measurements without compromising the estimation accuracy. This confirms the architecture versatility and potential in fostering the tuning and deployment of network measurement systems, revealing that a modular and configurable approach to sampling is a step forward for improving sampling scope and efficiency.

6 Conclusions

The present research work was focused on fostering the efficiency of network measurement systems through the development of a modular architecture for flexible deployment of packet sampling strategies. Supported by a consistent sampling taxonomy, the sampling-based measurement architecture was structured in three layers—*management plane*, *control plane* and *data plane*—covering the main elements involved in traffic measurements. Each layer was designed aiming at supporting the required compatibility with several measurement protocols and tools. The modular structure of the layers also provides the flexibility to accommodate mechanisms able to enhance the overall performance of current and forthcoming measurement needs. The architectural components were presented along with the different strategies and technologies that compose the several stages of a measurement process. A proof-of-concept was also provided, exploring the

flexibility of the proposed architecture when comparing the tradeoff between overhead and accuracy of representative sampling techniques in distinct measurement scenarios.

Although there is the lack of a study identifying the best sampling approach for each measurement goal and traffic scenario, the coverage of the literature and the outcomes from the present evaluation work have clearly demonstrated the need for a manifold solution. In this way, the resulting architecture, sustained by a modular taxonomy of sampling techniques, provides a valuable contribution to this research field. Having demonstrated the ability to implement several sampling techniques, we expect that the proposed architecture and framework will foster new comparative studies identifying the most suitable sampling techniques for the multitude of measurement scenarios where sampling has become mandatory.

In this way, future work intends to address an extensive and systematic comparative analysis toward various measurement tasks supported by packet sampling. In addition, we plan to take advantage of Software Defined Networking devices programability to implement the modular configuration of sampling techniques proposed along this work. This will endow commercial network devices with the ability to deploy a wide range of sampling techniques for efficient measuring of distinct network measurement tasks.

Acknowledgements This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT *Fundação para a Ciência e Tecnologia* within the Project Scope: UID/CEC/ 00319/2013.

References

1. Zseby, T., Molina, M., Duffield, N.: Sampling and Filtering Techniques for IP Packet Selection RFC 5475. Technical report, IETF. <http://datatracker.ietf.org/doc/rfc5475/> (2009)
2. Silva, J.M.C., Carvalho, P., Rito Lima, S.: Analysing traffic flows through sampling: a comparative study. In: 20th IEEE Symposium on Computers and Communication (ISCC), Cyprus (2015)
3. Jadwab, J., Phall, P., Pinna, B.: Traffic estimation for the largest sources on a network using packet sampling with limited storage. Technical report, Hewlett-Packard Laboratories, Bristol (1992)
4. Claffy, K.C., Polyzos, G.C., Braun, H.W.: Application of sampling methodologies to network traffic characterization, SIGCOMM. *Comput. Commun. Rev.* **23**(4), 194–203 (1993). doi:10.1145/167954.166256
5. Cozzani, I., Giordano, S.: Traffic sampling methods for end-to-end QoS evaluation in large heterogeneous networks. *Comput. Netw. ISDN Syst.* **30**(16–18), 1697–1706. <http://www.sciencedirect.com/science/article/pii/S0169755298001986> (1998)
6. Amer, P., Cassel, L.: Management of sampled real-time network measurements. In: Proceedings of 14th Conference on Local Computer Networks. IEEE Comput. Soc. Press, pp. 62–68. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=65244> (1989)
7. Tammaro, D., Valenti, S., Rossi, D., Pescapé, A.: Exploiting packet-sampling measurements for traffic characterization and classification. *Int. J. Netw. Manag.* **22**(6), 451–476 (2012). doi:10.1002/nem.1802
8. Duffield, N.: Fair sampling across network flow measurements. *ACM SIGMETRICS Perform. Eval. Rev.* **40**(1), 367 (2012). doi:<http://dl.acm.org/citation.cfm?id=2318857.2254800>
9. Hernandez, E.A., Chidester, M.C., George, A.D.: Adaptive sampling for network management. *J. Netw. Syst. Manag.* **9**(4), 409–434 (2001). doi:10.1023/A:1012980307500
10. Silva, J.M.C., Carvalho, P., Rito Lima, S.: A multiadaptive sampling technique for cost-effective network measurements. *Comput. Netw.* **57**(17), 3357–3369 (2013). doi:10.1016/j.comnet.2013.07.023

11. Duffield, N.G., Grossglauser, M.: Trajectory sampling for direct traffic observation. *ACM SIGCOMM Comput. Commun. Rev.* **30**(4), 271–282 (2000). doi:[10.1145/347057.347555](https://doi.org/10.1145/347057.347555)
12. Estan, C., Varghese, G.: New directions in traffic measurement and accounting. *SIGCOMM Comput. Commun. Rev.* **32**(4), 323–336 (2002). doi:[10.1145/964725.633056](https://doi.org/10.1145/964725.633056)
13. Singh, R., Kumar, H., Singla, R.K.: Analyzing statistical effect of sampling on network traffic dataset. In: Satapathy, S.C., Avadhani, P.S., Udgata, S.K., Lakshminarayana, S. (eds.) *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*. Springer International Publishing, pp. 401–408. http://link.springer.com/chapter/10.1007/978-3-319-03107-1_43 (2014)
14. Yang, L., Michailidis, G.: Sampled based estimation of network traffic flow characteristics. In: *IEEE INFOCOM 2007—26th IEEE International Conference on Computer Communications, (IEEE)* pp. 1775–1783. doi:[10.1109/INFCOM.2007.207](https://doi.org/10.1109/INFCOM.2007.207). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4215789> (2007)
15. Carela-Español, V., Barlet-Ros, P., Cabellos-Aparicio, A., Solé-Pareta, J.: Analysis of the impact of sampling on NetFlow traffic classification. *Comput. Netw.* **55**(5), 1083–1089 (2011). doi:[10.1016/j.comnet.2010.11.002](https://doi.org/10.1016/j.comnet.2010.11.002)
16. Lin, R., Li, O., Li, Q., Dai, K.: Exploiting adaptive packet-sampling measurements for multimedia traffic classification. *J. Commun.* **9**(12) (2014). <http://www.jocm.us/uploadfile/2014/1231/20141231030404520>
17. Kandula, S., Mahajan, R.: Sampling biases in network path measurements and what to do about it. In: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference IMC '09 (ACM, New York, NY, USA)*, pp. 156–169. doi:[10.1145/1644893.1644912](https://doi.org/10.1145/1644893.1644912) (2009)
18. Lee, M., Duffield, N., Kompella, R.: Two samples are enough: opportunistic flow-level latency estimation using NetFlow. In: *2010 Proceedings IEEE INFOCOM*, pp. 1–9. doi:[10.1109/INFCOM.2010.5462044](https://doi.org/10.1109/INFCOM.2010.5462044). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5462044> (2010)
19. Mahmood, A.N., Hu, J., Tari, Z., Leckie, C.: Critical infrastructure protection: resource efficient sampling to improve detection of less frequent patterns in network traffic. *J. Netw. Comput. Appl.* **33**(4), 491–502 (2010). <http://www.sciencedirect.com/science/article/B6WKB-4YBMFB6-1/2/9b91d8daa2364e0d025aed6088160da7>
20. Zhang, J., Luo, X., Perdisci, R., Gu, G., Lee, W., Feamster, N.: Boosting the scalability of botnet detection using adaptive traffic sampling. In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Ser. (ACM, New York, NY, USA), ASIACCS '11*, pp. 124–134. doi:[10.1145/1966913.1966930](https://doi.org/10.1145/1966913.1966930) (2011)
21. Huang, Y., Pullen, J.: Countering denial-of-service attacks using congestion triggered packet sampling and filtering. In: *Proceedings Tenth International Conference on Computer Communications and Networks (Cat. No. 01EX495), (IEEE)*, pp. 490–494 (2001). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=956309>
22. Brauckhoff, D., Tellenbach, B., Wagner, A., May, M., Lakhina, A.: Impact of packet sampling on anomaly detection metrics. In: *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, Ser. (ACM, New York, NY, USA) IMC '06*, pp. 159–164. doi:[10.1145/1177080.1177101](https://doi.org/10.1145/1177080.1177101) (2006)
23. Paredes-Oliva, I., Barlet-Ros, P., Solé-Pareta, J.: Portscan detection with sampled Netflow. In: *Traffic Monitoring and Analysis (Springer)*, pp. 26–33. http://link.springer.com/chapter/10.1007/978-3-642-01645-5_4 (2009)
24. Mai, J., Chuah, C.N., Sridharan, A., Ye, T., Zang, H.: Is sampled data sufficient for anomaly detection? In: *Proceedings of the 6th ACM SIGCOMM on Internet measurement—IMC'06, Ser. (ACM Press, New York, NY, USA) p. 165* (2006). <http://portal.acm.org/citation.cfm?doid=1177080.1177102>
25. Jae-Hyun, J., Cheol-Woong, A., Dongjoon, L., Sung-Ho, K.: DDoS attack detection using flow entropy and packet sampling on huge networks. In: *ICN 2014 : The Thirteenth International Conference on Networks (IARIA)*, pp. 183–190 (2014)
26. Zseby, T.: Deployment of sampling methods for SLA validation with non-intrusive measurements. In: *Proceedings of Passive and Active Measurements Conference (Fort Collins)* (2002)
27. Zseby, T.: Comparison of sampling methods for non-intrusive SLA validation. In: *Proceedings of the Second Workshop on End-to-End Monitoring Techniques and Services (E2EMon)* (2004)
28. Serral-Gracia, R., Cabellos-Aparicio, A., Domingo-Pascual, J.: Packet loss estimation using distributed adaptive sampling. In: *Network Operations and Management Symposium Workshops*, 2008.

- NOMS Workshops 2008. IEEE (IEEE), pp. 124–131 (2008). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4509938>
29. Sommers, J., Barford, P., Duffield, N., Ron, A.: Improving accuracy in end-to-end packet loss-measurement. In: Proceedings of the 2005 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications—SIGCOMM '05, (ACM Press, New York, New York, USA), vol. 35, p. 157 (2005). <http://dl.acm.org/citation.cfm?id=1080091.1080111>
 30. Dogman, A., Saatchi, R., Al-Khayatt, S.: An adaptive statistical sampling technique for computer-network traffic. In: 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP, 2010), pp. 479–483 (2010)
 31. Gu, Y., Breslau, L., Duffield, N., Sen, S.: On passive one-way loss measurements using sampledflow statistics. In: IEEE INFOCOM 2009—The 28th Conference on Computer Communications (IEEE), pp. 2946–2950 (2009). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5062264>
 32. Androulidakis, G., Chatzigiannakis, V., Papavassiliou, S.: Network anomaly detection and classification via opportunistic sampling. *IEEE Netw.* **23**(1), 6–12 (2009). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4804318>
 33. Choi, B.Y., Bhattacharyya, S.: Observations on cisco sampled ntfow. *ACM SIGMETRICS Perform. Eval. Rev.* **33**(3), p. 18 (2005). <http://portal.acm.org/citation.cfm?doid=1111572.1111579>
 34. Zseby, T., Hirsch, T., Claise, B.: Packet sampling for flow accounting: challenges and limitations. In: Claypool, M., Uhlig, S. (eds.) *Passive and Active Network Measurement*, Ser. *Lecture Notes in Computer Science*, vol. 4979, (Springer Berlin / Heidelberg), pp. 61–71 (2008). doi:[10.1007/978-3-540-79232-1_7](https://doi.org/10.1007/978-3-540-79232-1_7)
 35. Pescape, A., Rossi, D., Tammara, D., Valenti, S.: On the impact of sampling on traffic monitoring and analysis. In: 2010 22nd International Teletraffic Congress (ITC 22) (IEEE), pp. 1–8. (2010). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5608718>
 36. Chabchoub, Y., Fricker, C., Guillemin, F., Robert, P.: Deterministic versus probabilistic packet sampling in the internet. In: Mason, L., Drwiega, T., Yan, J. (eds.) *Managing Traffic Performance in Converged Networks*, *Lecture Notes in Computer Science*, vol. 4516, Springer, Berlin, Heidelberg, pp. 678–689 (2007). http://link.springer.com/chapter/10.1007/978-3-540-72990-7_60
 37. Castro, V., Carvalho, P., Lima, S.R., In: *A cooperative network monitoring overlay*. *Smart Spaces and Next Generation Wired/Wireless Networking*, Springer, pp. 475–486 (2011). http://link.springer.com/chapter/10.1007/978-3-642-22875-9_43
 38. Schad, J., Dittrich, J., Quiané-Ruiz, J.A.: Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proc. VLDB Endow.* **3**(1–2), 460–471 (2010). doi:[10.14778/1920841.1920902](https://doi.org/10.14778/1920841.1920902)
 39. Pras, A., Schoenwaelder, J.: On the difference between information models and data models—RFC 3444. Technical Report, IETF (2003). <https://datatracker.ietf.org/doc/rfc3444/>
 40. Claise, B., Trammell, B.: Specification of the IP Flow Information eXport (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011 (2013). <http://datatracker.ietf.org/doc/draft-ietf-ipfix-protocol-rfc5101bis/>
 41. Claise, B., Trammel, B.: Information Model for IP Flow Information Export (IPFIX)—RFC 7012. Technical Report IETF (2013). <https://datatracker.ietf.org/doc/rfc7012/>
 42. Dietz, T., Claise, B., Aitken, P., Dressler, F., Carle, G.: Information Model for Packet Sampling Exports. Technical Report, IETF RFC 5477 (2009). <https://datatracker.ietf.org/doc/rfc5477/>
 43. IP Flow Information Export (IPFIX): Entities (2015). <http://www.iana.org/assignments/ipfix/ipfix.xhtml>
 44. Dietz, T., Claise, B., Quittek, J.: Definitions of Managed Objects for Packet Sampling. RFC 6727 (2012). <http://datatracker.ietf.org/doc/rfc6727/>
 45. Case, J., Mundy, R., Partain, D., Stewart, B.: Introduction and Applicability Statements for Internet-Standard Management Framework—RFC 3410. Technical Report, IETF (2002). <https://datatracker.ietf.org/doc/rfc3410/>
 46. Aitken, P., Claise, B., McDowall, C., Schoenwaelder, J.: Exporting MIB Variables using the IPFIX Protocol draft-ietf-ipfix-mib-variable-export-09. Technical Report, IETF (2015). <https://datatracker.ietf.org/doc/draft-ietf-ipfix-mib-variable-export/>
 47. McCloghrie, K., Seligson, J., Reichmeyer, F., Smith, A., Sahita, R.: Structure of policy provisioning information (SPPI)—RFC 3159. Technical Report, IETF (2001). <https://datatracker.ietf.org/doc/rfc3159/>

48. UsLAR, M., Specht, M., Rohjans, S., Trefke, J., González, J.M.: The Common Information Model CIM: IEC 61968/61970 and 62325—A Practical Introduction to the CIM, vol. 66. Springer, New York (2012)
49. Silva, J.M.C., Carvalho, P., Rito Lima, S.: Enhancing traffic samplingscope and efficiency. In: 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (IEEE), pp. 71–72 (2013). <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6562848>
50. Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A.: Flow monitoring explained: from packet capture to data analysis With NetFlowand IPFIX. *IEEE Commun. Surv. Tutor.* **16**(4), 2037–2066 (2014). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6814316>
51. Claise, B., Johnson, A., Quittek, J.: Packet Sampling (PSAMP) Protocol Specifications. RFC 5476 (2009). <http://datatracker.ietf.org/doc/rfc5476/>
52. Orebaugh, A., Ramirez, G., Beale, J.: Wireshark and Ethereal Network Protocol Analyzer Toolkit. Syngress, Rockland (2006)
53. Jacobson, V., McCanne, S.: Lawrence Berkeley Laboratory, Berkeley, CA (2009)
54. Alcock, S., Lorier, P., Nelson, R.: ACM SIGCOMM Comput. Commun. Rev. **42**(2), 42 (2012). <http://dl.acm.org/citation.cfm?doid=2185376.2185382>
55. Silva, J.M.C., Carvalho, P., Lima, S.R.: Computational weight of network traffic samplingtechniques. In: 2014 IEEE Symposium on Computers and Communications (ISCC) (IEEE, Madeira, Portugal), pp. 1–6 (2014). <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6912467>
56. Shannon, C., Aben, E., Claffy, K., Andersen, D., Brownlee, N.: The CAIDA UCSD Anonymized Internet Traces 2008—equinix-chicago.dirA.20080430-170200.UTC.anon. Downloaded from http://www.caida.org/data/passive/passive_2008_dataset.xml (2008)
57. Shannon, C., Aben, E., Claffy, K., Andersen, D., Brownlee, N.: The CAIDA UCSD Anonymized Internet Traces 2014—equinix-chicago.dirA.20140619-131100.UTC.anon. Downloaded from http://www.caida.org/data/passive/passive_2014_dataset.xml (2014)
58. Krishnan, R., Yong, L., Ghanwani, A., So, N., Khasnabish, B.: Mechanisms for Optimizing Link Aggregation Group (LAG) and Equal-Cost Multipath (ECMP) Component Link Utilization in Networks—RFC 7424. Technical Report, IETF (2015). <https://datatracker.ietf.org/doc/rfc7424/>
59. Silverman, B.W.: Density Estimation for Statistics and Data Analysis, vol. 26. CRC Press, Boca Raton (1986)

João Marco C. Silva graduated in 2008 from University Federal of Sergipe, Brazil, received his M.Sc. and Ph.D. in Informatics from the University of Minho, Braga, Portugal, in 2011 and 2016, respectively. His research main interests include traffic sampling, network management, network monitoring, traffic modeling, quality of service, traffic classification and characterization.

Paulo Carvalho graduated in 1991 and received his Ph.D. degree in Computer Science from the University of Kent at Canterbury, United Kingdom, in 1997. He is currently Associate Professor in Computer Communications, Department of Informatics, University of Minho. His main research interests include broadband technologies, multiservice and mobile networks, and teletraffic analysis.

Solange Rito Lima graduated in 1992, received her M.Sc. and Ph.D. degrees in Computer Science from the University of Minho, Portugal, in 1997 and 2006, respectively. She is currently Assistant Professor in Computer Communications at Department of Informatics, University of Minho, and integrates the Board of Directors of this department as Vice Head-of-Department. Her research interests include multiservice networks and protocols, QoS/QoE, traffic control, monitoring issues in IP networks and emerging network technologies.