Towards a Holistic Semantic Support for Context-aware Network Monitoring: An ontology-based approach

Paulo Carvalho[1], Solange Rito Lima[1]
Luis Álvarez Sabucedo[2], Juan M. Santos Gago[2], and João Marco C. Silva[3]

[1] Centro Algoritmi, Universidade do Minho
4710-057 Braga, Portugal
{pmc,solange}@di.uminho.pt
[2] University of Vigo, Dept. of Telematics
Vigo, Spain
{Luis.Sabucedo,Juan.Gago}@det.uvigo.es
[3] HASLab, INESC TEC, Universidade do Minho
Braga, Portugal
joao.marco@inesctec.pt

Abstract. Monitoring current communication networks and services is an increasingly complex task as a result of a growth in the number and variety of components involved. Moreover, different perspectives on network monitoring and optimisation policies must be considered to meet context-dependent monitoring requirements. To face these demanding expectations, this article proposes a semantic-based approach to support the flexible configuration of context-aware network monitoring, where traffic sampling is used to improve efficiency. Thus, a semantic layer is proposed to provide with a standard and interoperable description of the elements, requirements and relevant features in the monitoring domain. On top of this description, semantic rules are applied to make decisions regarding monitoring and auditing policies in a proactive and context-aware manner. Use cases focusing on traffic accounting and traffic classification as monitoring tasks are also provided, demonstrating the expressiveness of the ontology and the contribution of smart SWRL rules for recommending optimised configuration profiles.

1 Introduction

Managing communication networks is a demanding multidimensional task requiring an increasingly number of support tools and expertise. This inherent difficulty stems from reasons such as the crescent heterogeneity in the type and capacity of network components, the plethora of services offered, and the different points of view regarding network planning, management and optimisation. Therefore, the search for more encompassing and versatile solutions to tackle the aforementioned issues is clearly recommended.

Network monitoring, as a fundamental task for assisting multiple network management areas (e.g., traffic accounting and classification, quality of service and security), should attend to each particular context where monitoring data and corresponding traffic measurements are required. In fact, a monitoring system can use relevant context information to provide customised and optimised monitoring facilities to meet both customers and network administrators needs.

Furthermore, context-aware monitoring may contribute for saving computational and communication resources, while allowing for the provision of more agile services. Therefore, the adoption of semantic-based approaches plays a key role, as well as, the use of traffic sampling techniques to reduce the amount of traffic collected, analysed and stored.

In this context, the authors propose a solution based on the use of semantic support and rules automatically applied to assist network monitoring. In particular, the proposal is endowed with properties considered mandatory when monitoring today's networks, namely, being context-aware, self-configurable and flexible, causing minimal interference with the normal network operation.

The first step in this process is to explore the current state of the art in this domain. In Section 2, a comprehensive review of related technologies is presented and previous work to tackle the above issues is discussed. From this study, it is clear that the potential of semantics, in a broad sense, has not yet been fully exploited in this domain. Although this technology has been applied to grant a layer indented to boost interoperatibility, many other possibilities remain unexplored. To fill this gap, a proposal for a context-aware monitoring architecture is presented using the support of semantics to automatise possibilities on monitoring policies. In section 3, the detailed description of the proposed model is presented, as well as, all relevant software components.

The next big step towards the proposed solution is the description of the ontology itself. In Section 4, the reader can find a description of the process to generate the ontology and also a description of the main classes and properties of the generated model. This model supports the description of all features and concepts considered relevant for the scope of this proposal and conveys the information required not just to describe but also to make decisions about network monitoring and auditing policies.

Facing the need to validate the formal model of the domain expressed by the above ontology, in Section 5 the reader can find the actual use of the proposed. By means of SPARQL Protocol and RDF Query Language (SPARQL) queries, it is shown how information can be obtained from the existing description. Examples of Semantic Web Rule Language (SWRL) rules are also presented for monitoring applied to traffic accounting and traffic classification. The rules drive decisions based on the actual status of the network and also pre-established behaviour directives from management staff. To enhance the flexibility of the model, these rules are adjusted by means of thresholds defined using automatic procedures.

The presented models are applied and checked using experimental data from a real network. In Section 6, the reader can find a description of this process. Finally, in Section 7, the main conclusions and lessons learnt are presented for future practitioners.

2 Related work

The problem of context-based network monitoring addressing specific management objectives on a tailored (per service) basis is not unexplored in the literature. Actually, this is a recurring problem attending to the ever growing

diversity of services supported and global usage, which may impact on the performance of communication networks.

To face the current demand of improved network monitoring in today's networks, strategies based on traffic sampling have been studied and deployed as an effective answer to cope with the diversity of services and high traffic volumes. These strategies have provided positive results in terms of characterising and classifying traffic [1–3], assuring SLA compliance [4, 5], performing Quality of Service (QoS) monitoring [6, 7], and protecting/securing the network [8, 9]. However, these approaches usually face serious issues as sampling solutions used to estimate a particular parameter correctly may not be adequate for a different parameter or traffic type [10], as consequence of traffic being heavily dynamic and heterogeneous. This requires having prior knowledge of the network traffic that can be used in a direct (or manual) way by the network administrators to tune the measurement processes according to the expected monitoring requirements. A distinct strategy from sampling-based approaches denominated In-Band Network Telemetry (INT) has been recently proposed for gathering multi-layer network information [11, 12]. INT framework follows a paradigm where network status is collected and reported resorting uniquely to the data plane, i.e., without intervention of the control plane. Although claimed to be generic, this approach poses several concerns and drawbacks as it implies a new protocolar layer, the presence of INT-capable devices, increasing the overhead of packets, as discussed in the literature [13]. Despite this, the emergence of a protocol-independent packet processor (P4) as a tool to support the management of INT messages in an interoperable and simple way [14] will possibly foster the adoption of INT for particular contexts, e.g. performance analysis in data centers.

Other alternative approaches have been explored to tackle context-based monitoring. These include the use of high-level expert systems based on ontological models. The idea on which these models are based is that the semantic expressiveness and interoperability introduced by this technology allows to overcome the difficulties mentioned. Among the works that follow this approach to map network managed objects into information models using semantic support are Structure of Management Information (SMI), Guidelines for Definition of Managed Objects (GMDO), Management Information Format (MIF), and IP Flow Information Export (IPFIX) [15, 16].

Considering network management activities, related research has explored ontological representation as a mechanism for supporting autonomic networks, in particular, for automated configuration. Among the first attempts to apply semantics in this domain, [17] must be acknowledge as an attempt to apply Web Ontology Language (OWL), OWL-S and Resource Description Framework (RDF) to tackle the issue from a non reactive point of view. The semantic support is not only a proper tool for supporting advanced services invoking. It is also a convenient solution to overcome interoperability issues. In [18], the authors take advantage of Natural Language Processing (NLP) frameworks to implement a solution to extract information from particular models and to integrate these pieces of information in a more open and interoperable description of the domain, i.e., an ontology.

In the literature, examples of more specific works which require traffic measurements focused mainly on QoS monitoring can be outlined, such as [19]. Also, in [20], a semantic model is proposed to describe the performance of Internet applications beyond the typical QoS metrics. Combining aspects such as the user profile, the requirements of each application and the network capabilities, a model to generate recommendations for a human user is devised.

Another increasingly relevant feature regarding network analysis is security from the perspective of the network as an infrastructure. In this domain, semantics has also a paramount role. In [21], the authors carry out an in-depth review of possible vulnerabilities and feasible attacks and provide a semantic model to describe them. This work could be the starting point of advanced support services to describe, publish and prevent attacks. Also, the interested reader can find in [22] an comprehensive review of the existing works concerned with providing, from different points of view, a semantic support for describing security concerns and features related to security on networks.

Although being considered a key enabler within network semantic management, and following [23], exploiting ontology's capabilities to face the challenges of selecting the most suitable sampling-based monitoring strategy in context-dependent network environments is still an open issue.

3 Context-aware monitoring architecture

3.1 Monitoring Architecture

The proposed context-aware monitoring architecture, represented in Figure 1, shows a high-level representation of the model designed.

At upper level, from a management plane perspective, each service or network management task should specify particular measuring requirements, which are expected to be subsequently satisfied. These requirements specificity will be handled within the control plane, where the expert recommendation system (detailed in Figure 2) will act to suggest downstream an adequate configuration profile for a set of selected measurement points (MPs). A configuration profile may specify, for instance, the most appropriate traffic sampling technique to use in MPs and the corresponding configuration parameters (see Section 6).

Therefore, depending on the monitoring context and network traffic variability, the system is expected to suggest a configuration profile so that the monitoring task can be efficiently accomplished. Efficiency is here understood as a trade-off between measurements accuracy and overhead.

3.2 Semantic recommender

The main modules of the semantic recommendation system are illustrated in Figure 2. As the reader may note, the *Inference Motor* is in charge of selecting a *Configuration Profile* adjusted to the measurement requirements. This profile is expected to provide the best possible configuration parameters and values according to the current state of the network and to the recommendation rules to apply within the present monitoring context. This software component may

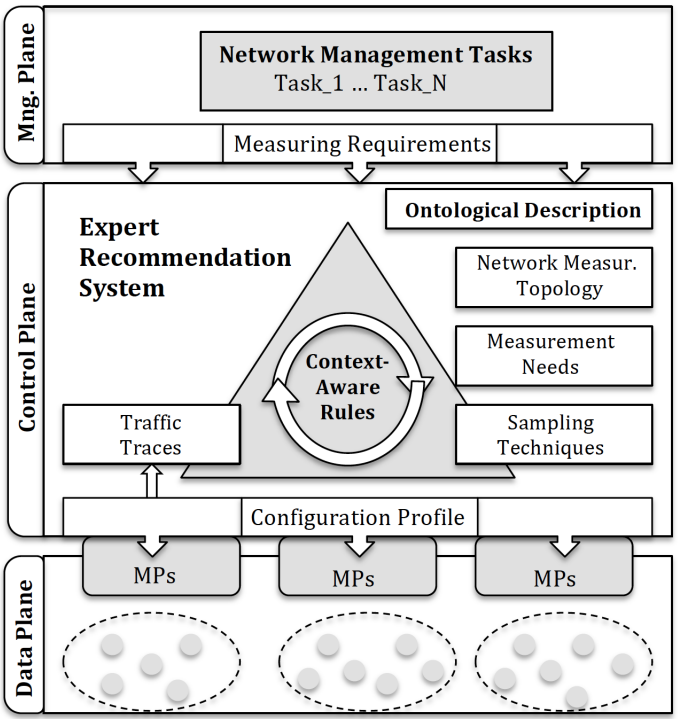

Fig. 1. Monitoring Architecture

also receive additional description of parameters from an external entity through a *Network Administration Profile*. This entity can be a human network operator specifying a particular management decision (administrative or technical) or an external software component such as an Software-Defined Network (SDN) controller.

The *Inference Motor* takes the *Knowledge Base* as main input. This component is a semantic information repository and comprises, mainly, two components:

- *ontology classes* - in this category, the definition of the classes required for the description of each element included within the model is provided. Rules expressed in SWRL are also considered as classes from a formal point of view.
- *ontology instances* - using the support of the former classes, instances containing the data about the current status of the system are provided. These instances contain information about the network topology, the traffic sampling techniques available, the network measurement needs and the current state of the network according to feedback provided by the traffic analyser.

One of the key elements in the knowledge repository is the set of rules specifying which changes must be applied in order to control the monitoring process and satisfy the ongoing measuring needs. These *Semantic Rules* are expressed in terms of the ontology using the support of SWRL (check section 5.4 for further details). The goal of these rules is to adjust and enact new configuration profiles according to parameters and thresholds that are determined by

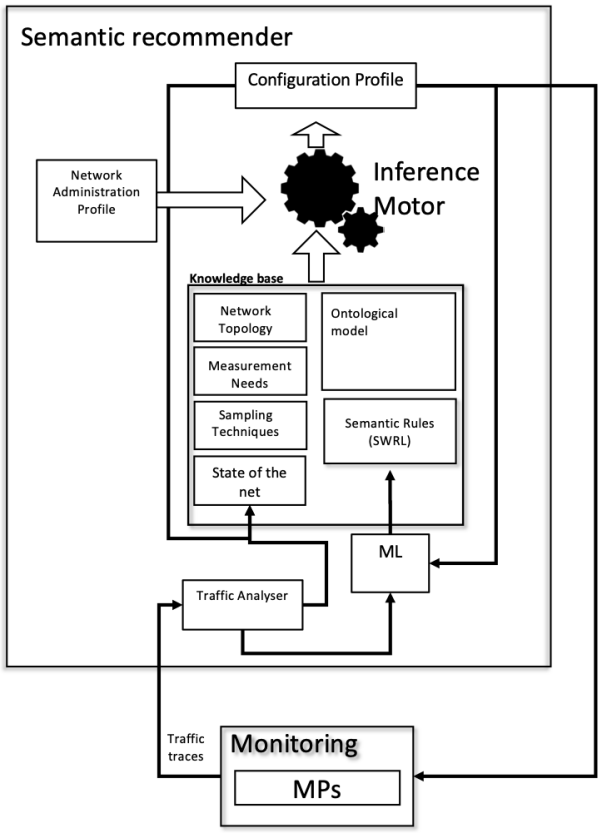

Fig. 2. Semantic Recommender

the *ML* module. This module incorporates Machine Learning (ML) algorithms for identifying traffic behaviours from real data. The *Traffic Analyser* module detects traffic fluctuations and assesses when these justify an adjustment to the monitoring configuration in place. The definition of these rules and models may vary from one particular network to another one, therefore, the participation of network administrators to make a final tune-up may be required.

4 Ontology definition

As the reader may expected, the proper definition of the semantic support plays a paramount role in the proposed monitoring architecture. Therefore, in order to ensure the feasibility of this long-term goal, a proper methodology for the semantic support definition, i.e., ontology, was selected bearing in mind the sustainability and maintainability of the final solution.

Ontology definition is not a new task in Knowledge Engineering. Actually, a number of methodologies for the definition of ontologies has been proposed in the state-of-the-art. Some of them must be acknowledged, such as those proposed in [24], [25] (known as the TOVE Methodology), [26] (known as Methontology), and the guide usually referred to as simple knowledge-engineering methodology [27] .

Recently, a new model to generate ontologies was proposed by Stuart [28]. This model presents a practical approach that explicitly pays attention to features such as sustainability and integration. This methodology, as the former ones, is an attempt to, somehow, support the engineering process of converting human knowledge into its formal representation that can be tackled by software agents. Even though this process is considered to be an iterative process, a set of concrete phases can be identified, namely: (a) Scope of the ontology; (b) Reuse of the ontology; (c) Identification of the appropriate software; (d) Acquisition of knowledge; (e) Identification of important terms; (f) Identification of additional terms; (g) Attributes and relationships; (h) Specification of definitions; (i) Integration with existing ontologies; (j) Implementation; (k) Evaluation; (l) Documentation; and (m) Sustainability.

As in any software-related process, the first step is linked to the definition of the scope of the final model. Within this step, several other steps are conducted. First of all, the application domain must be clearly set. In this case, the final purpose of the ontology is to serve as the basis for implementing a general-purpose sampling-based monitoring service. To define the requirements for pursuing this goal precisely, the identification of the so-called questions of competence, i.e., the questions to which the ontological system is expected to answer, needs to be addressed.

These questions arise from the interaction with the experts of the domain and are intended, as mentioned, to fix the aspects that should be contained in the proposed ontological model. Among the most significant ones, it is worth mentioning, for instance:

- Which MPs are border routers?
- Which is the available bandwidth in a certain MP?

– Which are the characteristics of a certain MP?
– Which are the current CPU load and memory usage within a MP?
– How many MPs compose the monitoring infrastructure (or overlay)?
– Which are the type and name of each MP, and the corresponding sampling techniques in use?
– Which are the active sampling technique and setup parameters at a particular MP?

Using these questions as the starting point, the process described in the methodology leads to the definition of a conceptual model (see Figure 3). Using this model, it is generated an ontology expressed in OWL and represented in Figure 4.

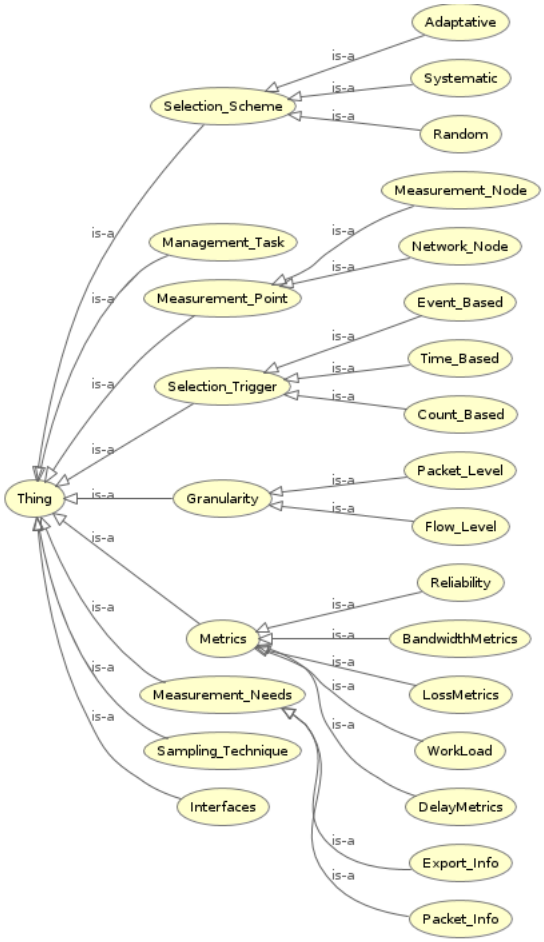

Fig. 3. Ontology specification

From the model presented, the reader can identify certain essential aspects in the characterisation of the reference domain. A number of fundamental classes are defined:

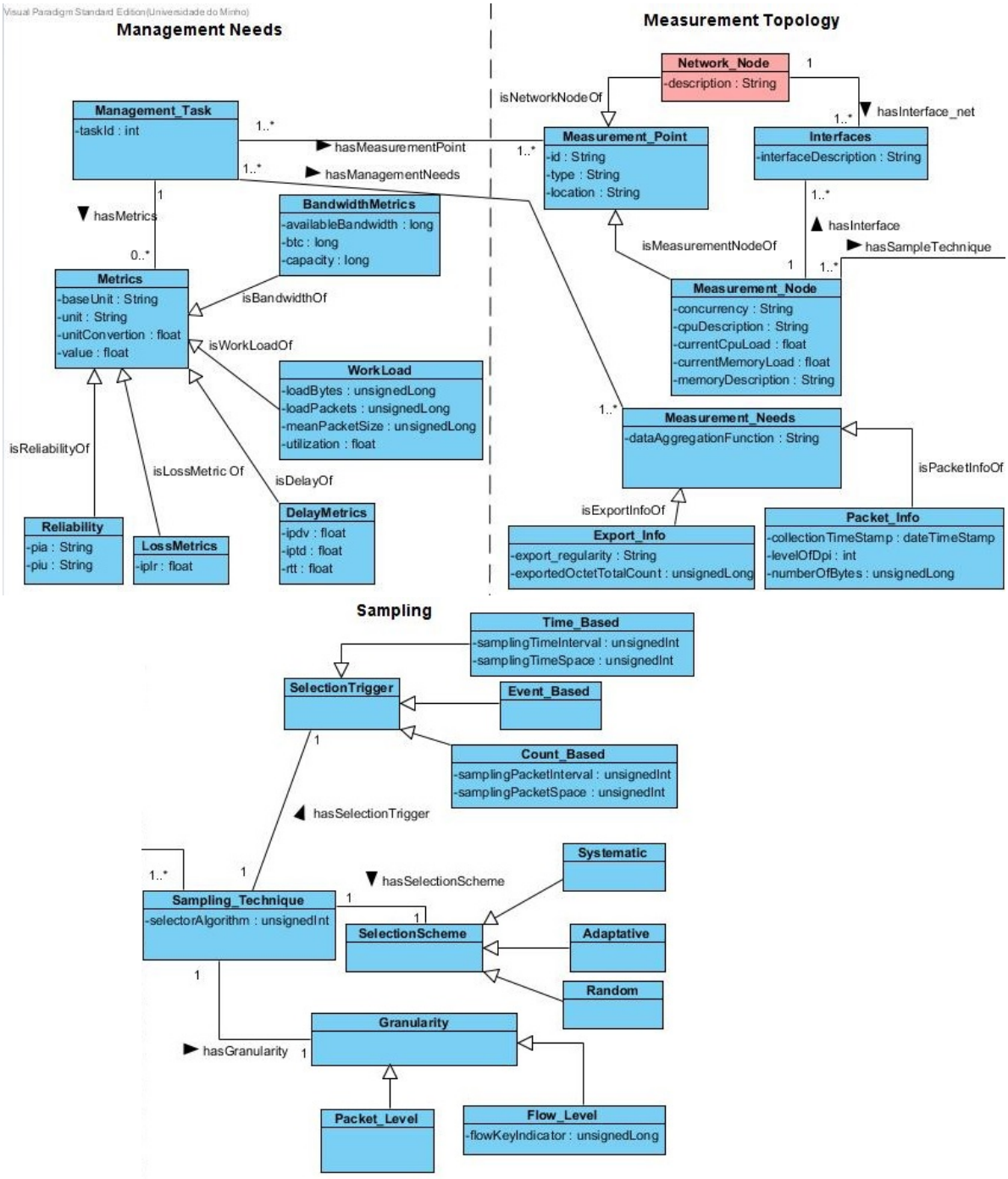

Fig. 4. Ontology: Model of classes

- Management Needs - this class supports the definition of management needs associated with tasks to be performed in the network. In this way, it is possible to declare which are the necessary quality and performance parameters at each moment. Each management task may need a particular measurement topology.
- Measurement Topology - this class supports the definition of the network elements involved in the measurement process. A measurement point has associated specific measurement nodes responsible for establishing traffic sampling-based measurements.
- Sampling - this class allows the definition of traffic sampling techniques according to their selection scheme, selection trigger and granularity.

On these classes, the instances (individuals in OWL terminology) to define the actual values are created. Semantic rules characterise the relationships between individuals from these classes. These rules assist the management of knowledge in a simpler and more interoperable way, as they are not actually coded in the final software, but included as semantic support expressed in OWL.

Using this semantic support, RDF statements can express that an individual from the class `Management_Task` (for instance, responsible for traffic accounting or traffic classification) uses an instance of the class `Measurement_Needs` to, indeed, declare its measurement needs. This statement requires the use of instances of the class `Metrics` to set/retrieve the corresponding measurement values.

The created ontological support includes a thorough definition of traffic sampling techniques, allowing to represent all representative techniques according to their characteristics. Actually, these techniques can be instantiated using the classes `SelectionScheme`, `SelectionTrigger` and `Granularity`, which determine the nature, the time and space characteristics regulating traffic sampling. The use of timers, packet counters or events allow to trigger the sampling process, regulating the intervals in which packets are collected. Depending on the existing load conditions in the network, the sampling intervals can be systematic, random or adaptive. More details about these aspects are presented on Section 6.

Classes also include data properties (or attributes in other terminologies) to include data on objects (e.g., `loadBytes` or `loadPackets`). Therefore, the system manager must instantiate (i.e., create an individual) of the class `Management_Task` and provide both, the object properties (such as `hasMeasurementPoint` and `taskID`) and the data properties to create in order to define a management task.

For the sake of brevity, not all stages in the methodology are described here, however, it is important to underline the validation of the semantic proposal. In this regard, competence questions play a crucial role both in the creation of an ontology, as they allow to justify its existence, and in the above-mentioned evaluation of the ontology. When creating the ontology, the questions of competence must be verified so that the development of the ontology does not deviate from the purpose initially defined. The reader can check this practical validation on the description of *use cases* within the following section.

5 Sematics in use

Once the semantic structure of the domain is established, the model is populated using current information from the University of Minho campus network. On the top of this model, a semantic engine will be set to take advantage of the knowledge already gathered. Thus, several services are deployed to maximise the utility of the contents available. These operations will include the tasks responsible for querying contents, creating new pieces of information, and exporting data for ML services in a smart way.

5.1 Querying the ontology

As mentioned in Section 4, the ontology must be able to answer a list of competence questions. These questions ground the existence of the ontology and allow to evaluate whether the ontology responds to the defined purposes. To achieve this, the SPARQL Protocol and RDF Query Language (SPARQL), based on the Resource Description Framework (RDF), are used.

As an example, a competence question from Section 4 is included. Other queries are covered within the traffic accounting case study discussed below. In Query1, a variable of the type `Measurement_Point` (line 4) is created, being its `id` assigned to the variable `name` (line 5). A variable `t` is also created, receiving the value of the `type` attribute associated with each `Measurement_Point` (line 6). In line 7, a filtering of all `type` attributes, whose content is equal to the expression"Border Router" is applied. The attributes `name` and `t` are select for output. The answer to this question returns MPs 1,2 and 5 as being borders routers. The output is not included here, as it can also be verified in the output of Query3.

- *Query1*: Which MPs are border routers?

  Input for Query1:

```
1  PREFIX ...
2  SELECT ?name ?t
3  WHERE {
4  ?mp rdf:type sm:Measurement_Point.
5  ?mp sm:id ?name.
6  ?mp sm:type ?t.
7  FILTER REGEX (?t,"Border Router").
8  }
```

5.2 Traffic accounting

For service providers, one of the costliest tasks in terms of time and resources is traffic accounting. However, this task is of vital importance in current commercial networks as the fulfilment of service agreements depend on it. For this reason, the use of ontological support was considered for assisting the control of traffic generated by users, their sessions duration, traffic rates and volumes, and type of services in use.

Accounting traffic involves several aspects that must be beard in mind. In Figure 5, the reader can see how the measuring requirements, the MPs, and the

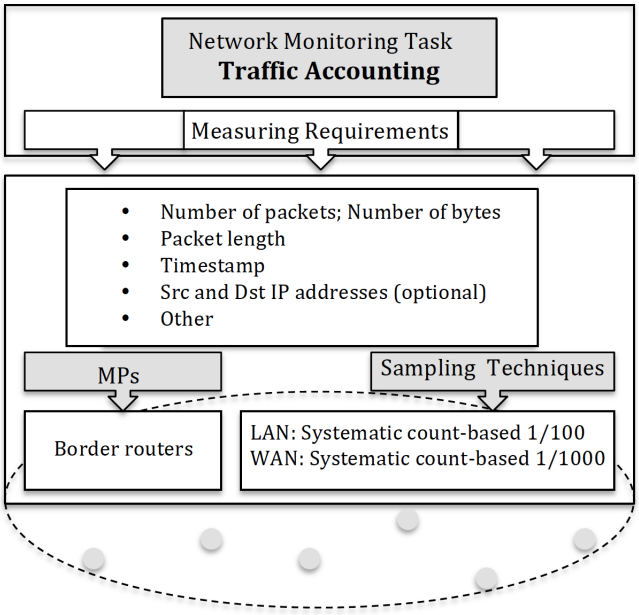

Fig. 5. Accounting task specification

most convenient sampling techniques are related in this domain. It is important to note that measurements accuracy and processing overhead (computational and storage) cannot be maximised simultaneously, and, therefore, a balance must be reached to obtain an accurate view of the network without interfering with normal network operation. Usually, to carry out tasks related to traffic accounting correctly, information regarding several parameters must be collected, such as the amount of traffic monitored (in packets and bytes); the size of the collected packets; the time instant when a sample is collected; the source and destination IP addresses; or other specific header (or payload) fields. Many decisions have to be made by network administrators to make a proper balance of the resources devoted to each task. In particular, it must be decided which are the most convenient MPs to run these tasks and the configuration of sampling techniques to use. Although the most common decision is to assign this role to border routers, distinct MPs can be used, as defined in Section 4.

Several features, such as memory and CPU available at a MP, or the amount of data to be collected, must be considered to decide which is the most convenient sampling technique to apply. In Figure 6, it is shown how specific parameters of an MP and the frequency of sampling are related, as experimental comparison reveals [29]. Experimental results suggest that a systematic count-based (SystC) technique on a high workload scenario may generate a significant impact on traffic volume and on CPU load. The same applies to memory occupancy, with a less pronounced ratio. This may suggest that the recommended sampling techniques for traffic accounting are SystC 1/100 for local area networks, and SystC 1/1000 for wide area networks[4].

[4] The parameter in SystC indicates that one packet is collected each one hundred or one thousand packets, respectively

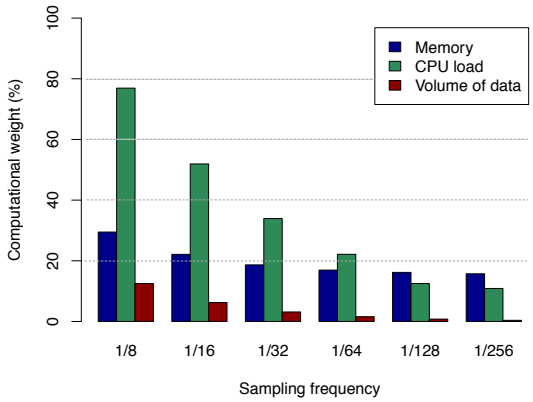

Fig. 6. Computational weight for different systematic sampling frequencies.

Within this context, two queries (Query2 and Query3) were formulated and the results obtained when performing traffic accounting using sampling techniques are presented below.

In the example of Query 2, six variables are created: `mp` of type Measurement_Point (line 4); `mn` of type Measurement_Node, being the `mn` value returned by the `hasMeasurementNode` object property associated with each Measurement_Point (lines 5, 7); `name` that collects the values of the `id` attribute associated with each Measurement_Point (line 6); `memory` and `cpu` that collects the values from the attributes `currentMemoryLoad` and `currentCpuLoad` associated with a `mn` (lines 8,9); `concurrency` which returns the status of the node regarding being dedicated to a single measurement task, or shared among multiple tasks (line 10). The variables selected for output are `name`, `memory`, `cpu` and `concurrency` (line 2). Figure 7 presents the results obtained for this query.

– *Query2*: Identify which measurement nodes are dedicated devices, and assess their current CPU load and memory usage.

Input for Query2:

```
1  PREFIX ...
2  SELECT ?name ?memory ?cpu ?concurren
3  WHERE {
4  ?mp rdf:type sm:Measurement_Point .
5  ?mn rdf:type sm:Measurement_Node .
6  ?mp sm:id ?name .
7  ?mp sm:hasMeasurementNode ?mn .
8  ?mn sm:currentMemoryLoad ?memory .
9  ?mn sm:currentCpuLoad ?cpu .
10 ?mn sm:concurrency ?concurren
11 }
```

In Query3, four variables are created: `mp` of type `Measurement_Point` (line 4); `name` that collects the values of the `id` attribute of each `Measurement_Point` (line 5); `technique` of type `Sampling_Technique` that collects the techniques associated to `Measurement_Points` through the object property `hasSampleTechnique`

name	memory	cpu	concurrency
"Point1"@	"76566"^^<http://www.w3.org/	"5.03"^^<http	"dedicated"@
"Point3"@	"86163"^^<http://www.w3.org/	"18.26"^^<htt	"dedicated"@
"Point5"@	"85551"^^<http://www.w3.org/	"97.27"^^<htt	"dedicated"@
"Point2"@	"96410"^^<http://www.w3.org/	"17.95"^^<htt	"shared"@
"Point4"@	"80765"^^<http://www.w3.org/	"10.76"^^<htt	"dedicated"@

Fig. 7. Output for Query2: Status of measurement nodes.

(line 6); and `type` that collects the types of `Measurement_Points` through `type` attribute (line 7). The variables selected for output are `name`, `technique` and `type` (line 2). Figure 8 shows the obtained values of this query.

– *Query3*: Identify MPs types, names and sampling techniques in use.

Input for Query3:

```
1 PREFIX ...
2 SELECT ?name ?technique ?type
3 WHERE {
4 ?mp rdf:type sm:Measurement_Point .
5 ?mp sm:id ?name .
6 ?mp sm:hasSampleTechnique ?technique .
7 ?mp sm:type ?type
8 }
```

name	technique	type
"Point4"@	MUST	"core"@
"Point2"@	SystT	"Border Router"@
"Point1"@	SystC	"Border Router"@
"Point3"@	RandC	"core"@
"Point5"@	LP	"Border Router"@

Fig. 8. Output for Query3 - MPs and corresponding sampling techniques in use.

5.3 Traffic classification

Traffic classification is a monitoring task that consists in analysing the traffic circulating in the network in order to assess which type of applications are in use and from which devices. Despite the simplicity of the concept, aspects such as the dynamic use of transport ports, the encapsulation of applications on HTTP, and the use of encryption protocols (IPsec, TTLS, etc.) reduce the efficiency of conventional classification strategies based on packet header fields. This has motivated the development of different and innovative approaches for traffic classification, such as based on flow or host behaviour. Despite that, in operational networks, due to its simplicity, it is still common to configure, for instance, firewalls rules based on transport level ports and IP addresses.

The main measuring requirements of traffic classification are depicted in Figure 9. Apart from timestamps, the level of packet inspection required from this monitoring task determines the number of header fields collected and, therefore, the number of bytes. As an example, a port-based classification strategy falls

within the level 4 of the protocol stack as it includes the source and destination ports in the collected traffic.

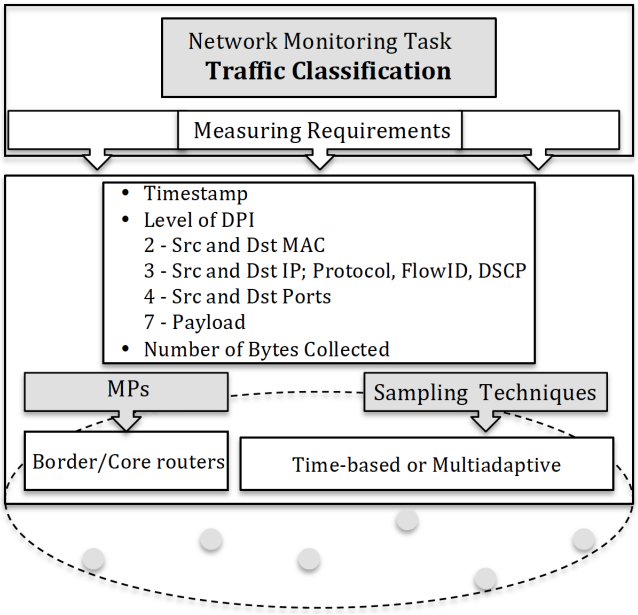

Fig. 9. Traffic classification task

When considering the use of traffic sampling to sustain this task with reduced amount of traffic, the aspects identified in Section 5.2 for Traffic Accounting need to be taken into account once again. According to [29], [30] and Figure 10, comparing the relation between the volume of data acquired and the number of different flows identified in the network, the most suitable sampling techniques for traffic classification are time-based and multiadaptive sampling (i.e., SystT and MuST). As expected, the techniques that sample larger volumes of data, identify a larger percentage of flows. However, when comparing count-based/random and time-based sampling involving similar data volumes, time-based techniques reveal to be more effective. This is explained by the intrinsic nature of time-based techniques in capturing successive packets during a sampling interval, improving the ability of capturing one or more packet of existing flows. Note, however, that these techniques are more demanding on computational resources from the network nodes were they are carried out (see details in Table 1). Therefore, the available levels of CPU and memory in MPs are aspects to consider when defining the technique to apply and corresponding parameters.

The model of classes of the ontology proposed in this work includes specific classes ($Packet\_Info$) to allow data collection for assessing traffic classification in the developed ontological system. The methodology used for questioning and obtaining answers from the ontology on traffic classification is similar to the traffic accounting case study, i.e., SPARQL is used to interrogate the system regarding traffic classification.

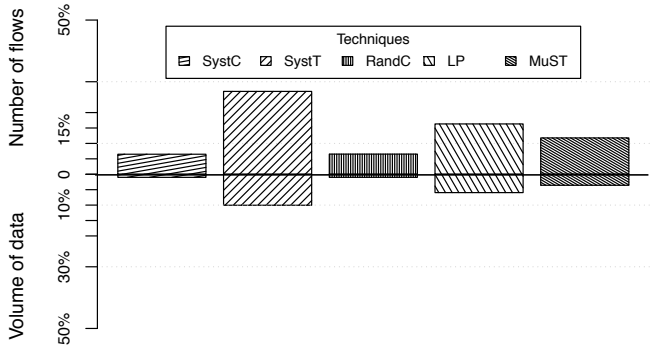

Fig. 10. Flow identification per volume of data acquired [30], for systematic (SystC,SystT), random (RandC) and adaptive (LP, MuST) techniques.

5.4 SWRL application

As mentioned in Section 3.2, the ontology within the system includes rules to assist network management tasks. For each task, these rules, expressed in SWRL, aim at grounding the decision making process regarding the adjustment of monitoring strategies and respective configuration parameters, whenever the network context changes significantly (either in traffic workload or MP computational resources).

The semantic rules defined are based on the concepts included in the ontology and represent behaviour, not on current state. The following examples illustrate specific SWRL rules for the monitoring tasks considered in the sections above.

Lets assume that for Task1 - Traffic Accounting, the primary objective is performing traffic accounting at the exit of an ISP domain (e.g., an egress router) so that the Service Level Agreements (SLAs) with downstream transit providers are not violated. A possible rule for determining the monitoring behaviour under high computational load (CPU/memory) could be:

SWRL rules for Task1

```
1     Management_Task(?t) ^
2     ?t rdf:type  :Accounting ^
3     hasMeasurementPoint(?t,?mp) ^
4     type(?mp,?tT) ^
5     swrlb:contains(?tT, "Egress") ^
6     currentCpuLoad(?mp, ?x) ^
7     swrlb:greaterThan(?x, 0.6) ^
8     currentMemoryLoad(?mp, ?y) ^
9     swrlb:greaterThan(?y, 0.75)
10        -> sampling_Technique(?mp,:SystC1)
```

Note that the use of certain thresholds (see lines 7 and 9) help deciding when to trigger a possible change in the monitoring configuration as an answer to a context change. As a matter of fact, these values are set based on the ML Module output shown in Figure 2. Therefore, according to the past and the current dynamics of system, these values can be adjusted for achieving a better monitoring performance. In line 10, $:SystC1$ corresponds to an individual of the ontology describing a specific monitoring configuration, in this case, the sampling technique Systematic Count-based with a parameter value 1/1000.

Lets now assume that for Task2 - Traffic Classification, the primary objective is performing traffic classification at the entrance of an ISP domain (e.g., an ingress router) for detecting anomalous or suspicious traffic flows. A possible rule for supporting flow classification at flow level could be:

SWRL rules for Task2

```
1   Management_Task(?t) ^
2   ?t rdf:type :Classification ^
3   hasMeasurementPoint(?t,?mp) ^
4   type(?mp, ?tT) ^
5   swrlb:contains(?tT, "Ingress") ^
6   currentCpuLoad(?mp, ?x) ^
7   swrlb:lessThan(?x, 0.4) ^
8   hasSamplingTechnique(?mp,?st) ^
9   hasGranularity(?st,?flow) ^
10  ?flow rdf:type :flow-level
11      -> sampling_Technique(?p,:MuST1)
```

Once the semantic layer defined as proposed is applied to all nodes in the network under management, monitoring the entire network would be simplified. The provision of an interoperable layer (common ontology), would benefit the development of network monitoring solutions in an objective, systematic and automatic manner.

6 Experimental results

The described model was tested under controlled conditions on a medium size network, actually, in a section of the University of Minho campus network. In this context, the system was deployed and experimental results obtained. As a matter of fact, results from testing systematic, random and adaptive sampling techniques under different measurement scopes and network status were acquired to derive the rules and models presented. The goal of capturing this data was to provide a realistic knowledge base to support the recommendation system. Table 1 presents a comparative performance analysis of different sampling techniques in three workload periods (low, moderate and high) in the network backbone of Gualtar campus along a typical workday. All measurements were performed resorting to simple and inexpensive single-board computers. Actually, Raspberry Pi Model B v.1 were used under a lightweight monitoring system research initiative. The results show the average use of computational resources (CPU and Memory in %) and the Relative Mean Error (RME) of each sampling technique when used to estimate instantaneous link usage (commonly applied in traffic accounting).

The experimental results evince that the selection and configuration of sampling techniques depend on: (i) the monitoring task considered (e.g., traffic accounting, classification); (ii) the different load conditions (e.g., link load and CPU usage); and/or (iii) MP location (border or core). Based on this premise, a baseline set of recommendations is presented in Table 2. For different network capacities and conditions, the qualitative metrics *High* and *Low* measuring the Link Load and CPU Usage, are defined according to predefined thresholds determined by internal policies in the network domain.

Table 1. Average use of computational resources and measurement accuracy

Technique	**High**			**Moderate**			**Low**		
	Mem	CPU	RME	Mem	CPU	RME	Mem	CPU	RME
SystC 1/8	29.5	76.9	0.0002	26.3	42.5	0.0012	17.1	11.9	0.007
SystC 1/100	16.4	14.9	0.0016	16.2	10.8	0.0097	15.4	5.0	0.025
SystT	18.2	20.1	0.038	19.4	17.9	0.030	19.3	14.6	0.046
RandC	17.3	18.3	0.008	16.9	16.9	0.0008	16.3	5.5	0.015
LP	17.2	97.3	0.11	17.6	96.7	0.13	16.6	27.4	0.05
MuST	16.2	10.8	0.13	16.9	10.7	0.003	17.1	8.8	0.009

Table 2. Sampling recommendations

Measurement scope and status					**Configuration**	
Task	**MP loc.**	**Link usage**	**CPU/Mem**		**Technique**	**Parameters**
Accounting	Border	Low	Low	->	RandC	1/100
			High	->	SystC	1/1000
		High	Low	->	RandC	1/100
			High	->	SystC	1/1000
	Core	Low	Low	->	MuST	200/500
			High	->	RandC	1/1000
		High	Low	->	SystC	1/32
			High	->	SystC	1/1000
Classification	Border	Low	Low	->	MuST	200/300
			High	->	SystC	1/100
		High	Low	->	MuST	200/500
			High	->	SystC	1/1000
	Core	Low	Low	->	LP	200/300
			High	->	RandC	1/100
		High	Low	->	MuST	200/500
			High	->	SystC	1/1000

In our case, using these rules for fine tuning the recommendations, a noticeable improvement regarding network performance analysis can be experienced. In Figure 11, it is shown this impact in terms of the ability of assisting flow analysis, i.e., accurately identifying the total number of unidirectional flows (%flows) and heavy hitter flows (%HH)[5] in the network. In particular, it is noticeable that the recommendation of using SystC and RandC sampling techniques for *Classification* in high CPU/Memory usage conditions results in a substantial reduction on the number of packets collected and processed, without impacting on the %HH flows identified. For the remaining flows, the ratio between the packets processed and the % of flows identified is still advantageous. As the reader may note, the rules proposed should be adjusted to each network and its particular needs and constrains. Nevertheless, the mechanism for automatic detecting the triggering context and the decision support for the most suitable response turn out in a suitable tool for managing the network performance.

7 Conclusions

In the authors' opinion, challenges and problems in finding the most adequate options for assisting network management tasks in a more versatile and cus-

[5] The notion of heavy hitter refers to 20% of the largest flows in terms of number of packets).

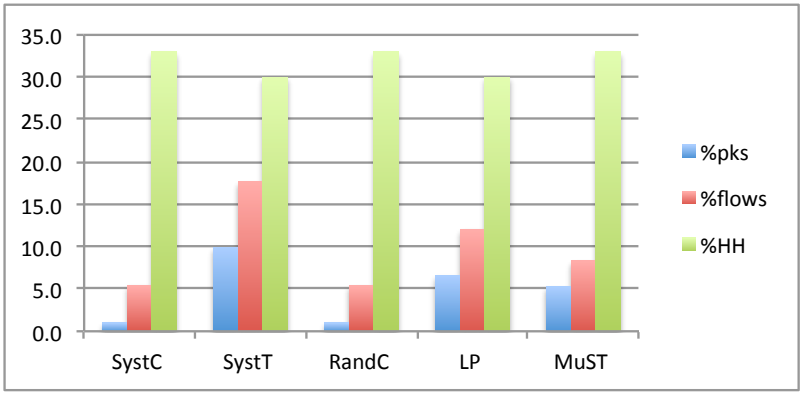

Fig. 11. Performance results

tomisable way will be a constant in the coming years. In this trend, in recent years different paradigms have been proposed, exploring new models and heuristics [31].To tackle this problem, this work has proposed the construction of a semantic model to assist the development of solutions based on expert agents for context-aware network monitoring. That is why an ontology has been proposed to describe the domain and a software architecture designed providing these high added-value services for network monitoring in a highly automated way.

The proposed model was validated by verifying competency questions and setting up SWRL rules for supporting of two relevant monitoring tasks - *traffic accounting* and *traffic classification* in today's ISP networks that can be used by practitioners for further attempts. Also, experimental tests were conducted on a controlled test scenario. Therefore, even a validation on a real scenario is still pending, tests carried out in a controlled scenario and the formal validation of the semantic model show the feasibility and potential of the presented approach.

Using the know-how obtained in this work, the authors are planning to extend the support of the system not just to perform the traffic accounting and classification tasks, but also to trigger additional reactive measures. In this way, rules to change either the monitoring topology according to particular network events or administrative policies regarding traffic conditioning can be launched in an automatic manner, as response to network dynamics. The authors also evaluate the possibility of taking advantage of INT framework and P4 language to make the most of the proposed semantic model.

Acknowledgements This work has been funded by EU H2020 - The EU Framework Programme for Research and Innovation 2014-2020, grant agreement No. 732505, and by FCT – Fundação para a Ciência e Tecnologia – within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Lin, R., Li, O., Li, Q., Dai, K.: Exploiting adaptive packet-sampling measurements for multimedia traffic classification. Journal of Communications **9**(12) (2014)

2. Tammaro, D., Valenti, S., Rossi, D., Pescapé, A.: Exploiting packet-sampling measurements for traffic characterization and classification. International Journal of Network Management **22**(6) (2012) 451–476
3. Silva, J.M.C., Carvalho, P., Lima, S.R.: Inside packet sampling techniques: exploring modularity to enhance network measurements. International Journal of Communication Systems **30**(6) (2017)
4. Zseby, T., Hirsch, T., Claise, B.: Packet sampling for flow accounting: Challenges and limitations. In: International Conference on Passive and Active Network Measurement, Springer (2008) 61–71
5. Hu, C., Wang, S., Tian, J., Liu, B., Cheng, Y., Chen, Y.: Accurate and efficient traffic monitoring using adaptive non-linear sampling method. In: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, IEEE (2008) 26–30
6. 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. Journal of Network and Computer Applications **33**(4) (2010) 491–502
7. Gu, Y., Breslau, L., Duffield, N., Sen, S.: On passive one-way loss measurements using sampled flow statistics. In: INFOCOM 2009, IEEE, IEEE (2009) 2946–2950
8. Yoon, S., Ha, T., Kim, S., Lim, H.: Scalable traffic sampling using centrality measure on SDNs. IEEE Communications Magazine **55**(7) (2017) 43–49
9. Jun, J.H., Ahn, C.W., Kim, S.H.: Ddos attack detection by using packet sampling and flow features. In: proceedings of the 29th annual ACM symposium on applied computing, ACM (2014) 711–712
10. Duffield, N., et al.: Sampling for passive internet measurement: A review. Statistical Science **19**(3) (2004) 472–498
11. Kim, C., Sivaraman, A., Katta, N.P., Bas, A., Dixit, A., Wobker, L.J.: (In-band network telemetry via programmable dataplanes)
12. Vestin, J., Kassler, A., Bhamare, D., Grinnemo, K., Andersson, J., Pongracz, G.: Programmable event detection for in-band network telemetry. In: 2019 IEEE 8th International Conference on Cloud Networking (CloudNet). (2019) 1–6
13. Tang, S., Li, D., Niu, B., Peng, J., Zhu, Z.: Sel-int: A runtime-programmable selective in-band network telemetry system. IEEE Transactions on Network and Service Management (2019) 1–1
14. Bhamare, D., Kassler, A., Vestin, J., Khoshkholghi, M.A., Taheri, J.: Intopt: In-band network telemetry optimization for nfv service chain monitoring. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). (2019) 1–7
15. Martinez, A., Yannuzzi, M., López, V., López, D., Ramírez, W., Serral-Gracià, R., Masip-Bruin, X., Maciejewski, M., Altmann, J.: Network management challenges and trends in multi-layer and multi-vendor settings for carrier-grade networks. IEEE Communications Surveys & Tutorials **16**(4) (2014) 2207–2230
16. Wong, A.K.Y., Ray, P., Parameswaran, N., Strassner, J.: Ontology mapping for the interoperability problem in network management. IEEE Journal on selected areas in Communications **23**(10) (2005) 2058–2068
17. Xu, H., Xiao, D.: Applying semantic web services to automate network management. In: Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on, IEEE (2007) 461–466
18. Martinez, A., Yannuzzi, M., de Vergara, J.L., Serral-Gracià, R., Ramírez, W.: An Ontology-Based Information Extraction System for bridging the configuration gap in hybrid SDN environments. In: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, IEEE (2015) 441–449
19. Rodrigues, C., Lima, S.R., Sabucedo, L.M.Á., Carvalho, P.: An ontology for managing network services quality. Expert Systems with App. **39**(9) (2012) 7938–7946
20. Moraes, P.S., Sampaio, L.N., Monteiro, J.A., Portnoi, M.: Mononto: A domain ontology for network monitoring and recommendation for advanced internet applications users. In: Network Operations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE, IEEE (2008) 116–123
21. Simmonds, A., Sandilands, P., Van Ekert, L.: An ontology for network security attacks. In: Asian Applied Computing Conference, Springer (2004) 317–323

22. Silva, D.V., Rafael, G.R.: Ontologies for network security and future challenges. In: International Conference on Cyber Warfare and Security, Academic Conferences International Limited (2017) 541
23. Silva, R.F., Carvalho, P., Rito Lima, S., Álvarez Sabucedo, L., Santos Gago, J.M., Silva, J.M.C.: An ontology-based recommendation system for context-aware network monitoring. In Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S., eds.: New Knowledge in Information Systems and Technologies, Cham, Springer International Publishing (2019) 373–384
24. Uschold, M., King, M.: Towards a methodology for building ontologies. In: In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95. (1995)
25. Grüninger, M., Fox, M.: Methodology for the Design and Evaluation of Ontologies. In: IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, April 13, 1995. (1995)
26. Fernández-López, M., Gómez-Pérez, A, Juristo, N.: Methontology: From ontological art towards ontological engineering. Symposium on Ontological Art Towards Ontological Engineering of AAAI. (1997) 33–40
27. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical report (2001)
28. Stuart, D.: Practical Ontologies for Information Professionals. Facet Publishing (2016)
29. Silva, J.M.C., Carvalho, P., Lima, S.R.: Computational weight of network traffic sampling techniques. In: 2014 IEEE Symposium on Computers and Communications (ISCC), IEEE (2014) 1–6
30. Silva, J.M.C., Carvalho, P., Lima, S.R.: Analysing traffic flows through sampling: A comparative study. In: 2015 IEEE Symposium on Computers and Communication (ISCC). (2015) 341–346 ¡p¿n/a¡/p¿.
31. Bhamare, D., Krishnamoorthy, M., Gumaste, A.: Models and algorithms for centralized control planes to optimize control traffic overhead. Comput. Commun. **70**(C) (2015) 68–78