

VBaaS: VNF Benchmark-as-a-Service

Raphael Vicente Rosa and Christian Esteve Rothenberg
School of Electrical and Computer Engineering (FEEC)
University of Campinas (UNICAMP)
Email: {raphaelvrosa,chesteve}@dca.fee.unicamp.br

Robert Szabo
Ericsson Research
Budapest, Hungary
Email: robert.szabo@ericsson.com

Abstract—When rolling out Network Function Virtualization (NFV) services, resource monitoring becomes a critical task subject to different cost-accuracy tradeoffs depending on whether continuous monitoring or more static infrastructure resource views are taken. In this context, we propose Virtualized Network Functions (VNF) Benchmark-as-a-Service (VBaaS) to enable not only run-time resource evaluation but also test-before-deploy opportunities for VNFs and NFV Infrastructures. We describe the motivation behind VBaaS and its main value proposition for a number of use cases around the orchestration tasks of VNF Forwarding Graphs. We present the main components of VBaaS along their system interactions and interfaces, discussing the main benefits of adopting VBaaS and open research issues. Addressing the identified challenges and finalizing our proof of concept VBaaS are our main ongoing work activities.

I. INTRODUCTION

Network Function Virtualization (NFV) must cope with complex resource provisioning challenges to offer carrier-grade services over virtualized views of compute, storage and network domains, as outcomes from resource abstractions (e.g. topology hiding). Among the promises of NFV is to handle dynamic workloads by scaling up/down or in/out to meet Key Quality Indicator (KQI) requirements [1]. The goal to flexibly introduce value added services through Network Service Chains (NSCs) [2] puts further stress on orchestration decisions to ensure end-to-end service requirements over multiple NFV Infrastructure (NFVI) Points of Presence (PoPs) [3].

The logically centralized view at the Network Function Virtualization Orchestrator (NFVO) allows monitoring processes to gathering information from different PoPs well-suited to provide valuable observations of NFVI transient states. The effectiveness of embedding algorithms can significantly vary depending on the accuracy of the resource profiles for the Virtualized Network Functions (VNFs) in the VNF Forwarding Graph (VNF-FG) and the accuracy of the available virtualized resources at the different NFVI PoPs [4].

Currently, different infrastructure monitoring techniques are being employed to obtain consistent infrastructure status, including pooling, periodical status reports, trapping, pre-programmed triggered state changes of certain resources levels, and so on. Trade-offs exist whether centralized or distributed techniques are used and the related accuracy versus scalability concerns. Static views of infrastructure statuses for deploying VNFs can be inadequate or insufficient to offer certain performance guarantees that will assure customers KQIs needs.

Resource views vary due to traffic engineering, capacity planning techniques, changes in tenant dynamic workloads,

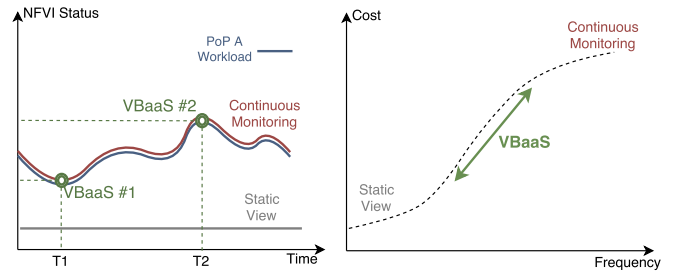


Fig. 1: VBaaS Motivation

or infrastructure failures. The underpinning infrastructure experiment variable resource consumption as VNF-FGs occupy distributed NFVIs with fragmented workloads driven by customers temporal requirements. Observability points at variable time scales can provide NFVO infrastructure statuses trade-offs when performing VNF-FGs embeddings –meaning possible inaccurate NFVI views resulting in higher costs. Consequently, VNFs may see their performance jeopardized depending on the resource-oriented inputs used at allocation time.

Aiming to provide a sweet spot solution, we present VNF Benchmarking as a Service (VBaaS) as a framework for NFVI PoPs/VNFs performance profile construction, which overcomes limitations of static views while avoiding continuous monitoring overheads, altogether providing lower complexity/cost and high flexibility when provisioning VNF-FGs and carrying service assurance processes (see Fig. 1). VBaaS is provided on-demand and can take place whenever precise infrastructure resources statuses are required. The main contribution aspects behind VBaaS are described below.

On-demand test-before-deploy: to avoid continuous monitoring overheads and inaccurate views of static approaches, VBaaS provides the option of consistent evaluations of different NFVI PoPs to offer VNF-FGs customized performance assessments for tenants and NFVO embedding algorithms. Sub-cases of VBaaS oriented decisions can be defined for migration/scaling and auditing VNF-FGs with parameters that provide assurance of capacity planning and network properties verification like reachability or isolation.

Conformance testing: VNF developers can build and adjust VBaaS profiles for their VNFs as required; infrastructure providers embracing Dev-Ops for VNFs continuous integration can define services compliance levels through different interoperable VBaaS profiles of VNFs; NFVI providers can verify regulatory requirements of shared infrastructures and define

service level compliance specifications as a way of proactive policy enforcement during infrastructure setups.

In the rest of the paper we first present related work (Sec. II) and then introduce the main aspects of VBaaS, including its components and general design guidelines (Section III). VBaaS use cases are then described (Sec. IV) followed by the analysis of our proposal (Sec. V) before concluding the paper with a glimpse on our ongoing/future work (Sec. VI).

II. RELATED WORK

Closest to our VBaaS activities is recent work [5] on VNFs benchmarking by adding internal instrumentation to fine tune NFVI resources execution. Motivated through two VNF use cases (vCDN and vWAN accelerator), the presented results point to less resource usage than expected/recommended by the VNF developers. Two important observations are discussed, firstly, the main goals of internal instrumentation of VNFs are justifiable by a richer set of benchmark metrics, the instrumentation before operational execution for fine-tuning and usage of the obtained results to set better monitoring points when VNFs are deployed. Secondly, a list of observed states that metrics depend on: number of layers instrumented and their respective configuration; a per-layer set of granularity measurements; and the sampling rate. Issues related with stability of the instrumentation framework, chained VNF, their analysis in commodity setups are considered for future work.

IETF/IRTF is another realm where a number of related efforts are going on. Work at the NFVRG [6] discusses the description of resources management for service chaining. The draft presents a series of use cases (e.g., fail-over, path and traffic optimization) to advocate for the optimization of the network services quality and NFVI resources usage. It defines some sub-actions (e.g., evaluate and replace VNF instances) which can be coordinated to improve NFV resources management with the observation of some metrics such as topological location, utilization rate, throughput, energy consumption, among others to scale-in/out or up/down VNF instances.

On recent related open source developments, Open Platform for NFV (OPNFV) is a project maintained by Linux Foundation which aims to develop an open source NFV platform based on ETSI requirements. Initially focusing on the Virtualized Infrastructure Manager (VIM) and its associated infrastructure resources layer, there are currently some project proposals [7] under the code names Yardstick and Transformer related to the extraction of information and testing of the data plane elements. Yardstick proposes to verify, via test cases and test stimuli, the infrastructure compliance when running VNF applications. Transformer focuses on related 3GPP standards for evaluation of Network Functions interoperability and functionality. In addition, two other projects, Doctor and Inspector, aim to respectively ensure the existence of auditing framework for NFV and provide data collection for failure prediction.

III. VBaaS: VNF BENCHMARK-AS-A-SERVICE

We now dive into VBaaS motivation points to provide an overall view behind its main components and interactions.

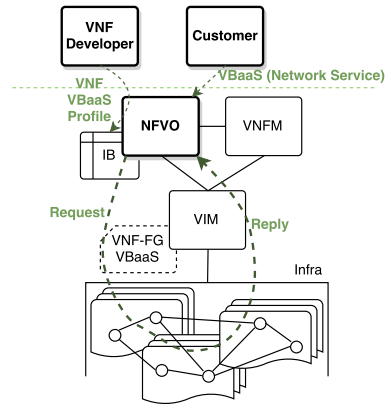


Fig. 2: VBaaS: VNF Benchmark-as-a-Service

A. General Aspects

Below, VBaaS big picture is presented via key questions about its main motivational factors and particular analysis.

Why VBaaS? Mainly justified by two NFV requirements: (i) performance reliability, and (ii) agility. With the premise of customers demands for infrastructure resources with predictable performance requirements, VBaaS contributes to service guarantees for VNF-FGs. As a side effect, VBaaS-like capabilities allow replication of VNFs/VNF-FGs towards plug-and-play deployments agnostic to infrastructure resources, e.g., performing all pre-(migration or scaling) verification steps before executing those tasks, allowing high-fidelity capacity planning and hence contributing to the networking agility of customers, service and infrastructure providers. The VBaaS model avoids the complexity of continuous monitoring tools with deep analysis inspections of infrastructure while providing a consistent view of current state when and where needed.

What is VBaaS? In essence it is a NFVI/VNF features extraction framework to construct useful performance profiles. The proposed framework pursues assembling and refining infrastructure and VNF benchmarks to fill VNF-FGs in accordance to service guarantees, which can be expressed via Key Performance Indicators (KPIs). VBaaS introduces methods to assist NFV orchestration decisions and aid both end-customers and VNF developers. The compromise taken is not going through the path of temporal series analysis via continuous monitoring processes but solving an on-demand request for tailor-made infrastructure information about NFVI capabilities for VNF-FGs allocation, scaling or even state verification with high flexibility at low cost/complexity.

How VBaaS can be performed? As it shall become clear through use case discussions (Sec. IV), VBaaS is per-se an independent as-a-Service framework that can be leveraged by multiple NFV parties. For instance, orchestrators looking for a benchmark-guided decision process for VNF-FG embedding and low complex resources' features extraction can leverage the VBaaS toolkit as necessary. VBaaS for VNFs can be instantiated as a tiny scale single instance or as a complex chain required for VNF-FGs evaluation. Using different VNFs and NFVI benchmarking profiles with varying features details, VBaaS is capable of delivering refined and correlated metrics

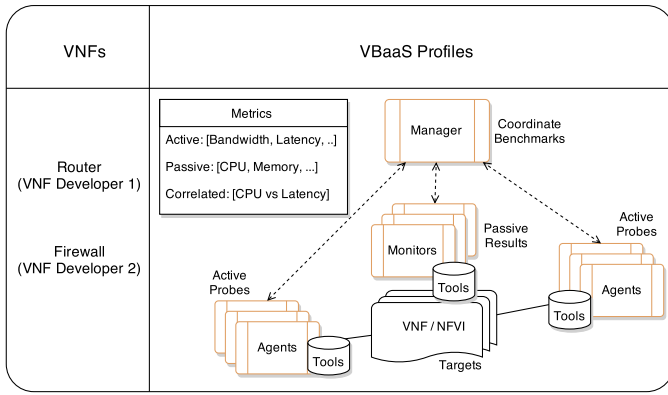


Fig. 3: Information Base

for orchestrator decisions consider them as resulting KPIs in test-before-deploy and certification processes.

When VBaaS can be used? Depending on orchestrator policies, VBaaS can provide capabilities to refine monitoring views at different time scales, and only present to customers the recommended NFVI PoPs locations best-suited for specific VNF-FG deployments, for instance. Considering the complexity of the entire VNF-FG embedding process (overhead, infrastructure stress, and overload over deployed VNFs/VNF-FGs), VBaaS adds operational costs and time. According to preferred orchestrator schedules and policies, these costs must be taken into account to not overload the NFVI neither break restrictions of VNF-FGs with embedding time priorities. Also, strategies for using VBaaS together with continuous monitoring processes can be evaluated and architected yielding better synchronized efforts depending on orchestrator and customer needs and workloads. Otherwise, VBaaS can be used whenever a new VNF/NFVI performance profile needs to be created.

B. Actors, Components and Interactions

Highlighted in Fig. 2 and described below along their respective functional behaviour, Customer, VNF Developer and NFVO are the main actors of VBaaS.

VNF Developer: an open interface is provided by the NFVO to VNF developers specify characteristics of their VNFs to create a custom profile with requirements and metrics upon which the VNF shall be benchmarked, referred herein VBaaS Profile as explained next. VBaaS Profiles are stored in a repository named Information Base (IB), maintained by the NFVO.

Customers: are actors consuming the VBaaS interfaces from the NFVO to request for analysis of candidate NFVI PoPs based on VNF targets. VNFs can be specified with possible intents for deployment representing constraints in the search space of NFVO embedding algorithms. Customers receive as response a profile containing VBaaS results with evaluations and analysis of performance guarantees for each one of the specified targets, i.e., NFVI PoPs.

NFVO: performs the interconnection between VNF Developers and the IB, storage of VBaaS profiles. NFVO attends VBaaS calls performing analysis on the benchmarking results to present customers a ranked list of NFVI PoP alternatives.

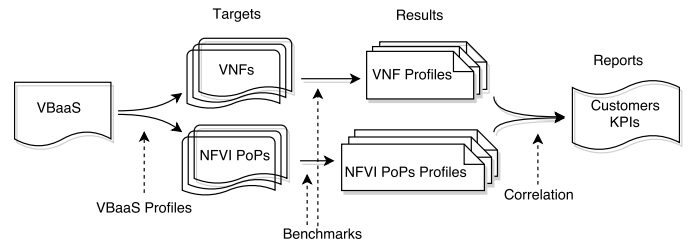


Fig. 4: VBaaS Profiles

Part of the process of NFVO is checking if correct definitions of VBaaS Profiles attend the desired properties of customers requests for NFVI resources. The same applies for VNFs benchmarks, when particular metrics apply to different infrastructure virtualization levels. Hence, as NFVO main duty is the construction and exposure of VBaaS abstractions and targets for consistent analysis and customers needs.

The Information Base (IB) (see Fig. 3) is a database to store and compose VBaaS Profiles. It serves as a ledger to the NFVO build VBaaS Service Graphs. It is inherited by IB the possibility to compose structures of nested VBaaS Profiles according to developers specifications (e.g., DPI composed by packets sniffer, parser and analyser VNFs).

VBaaS Profile is a set of agents, monitors, and managers – the main VNF components of VBaaS, detailed below– with a proper topology to evaluate the associated target (VNF or NFVI PoPs) containing the proper metrics expected as outputs along the respective benchmark tools, operations, and parameters (see Fig. 3). Figure 4 shows when NFVO uses IB to feed VBaaS profiles. Having VNFs and NFVI PoPs as targets different profile results are presented for each case. After correlating results, the customers’ KPIs can be presented as the final outputs of the VBaaS process.

In VBaaS, VIM provides infrastructure-level processes based on the VBaaS Service Graph sent by NFVO. VIM decomposes it in a series of VNFs instances (i.e., Agents, Monitors, and Managers) that execute the NFVI PoP or VNF benchmarks, and return the obtained metrics used to construct a VBaaS report to be sent back to the NFVO according to the profiles requested. The benchmarking VNFs have Application Programming Interfaces (APIs) to receive configuration parameters for benchmarking instances such as bandwidth or packet loss probes as described below.

Agent: executes active probes using benchmark tools to collect end-to-end metrics by interacting with other agents. For example, it can be used probe measurements of bandwidth and latency in a multi-point distributed topology. Agent’ APIs are open and extensible (e.g., to plug-and-bench new tools and metrics) and receive commands from the Manager for synchronization purposes (e.g., test duration/repetition).

Monitor: performs passive metrics monitoring and collection based on benchmarks evaluated by agents (e.g., monitor CPU utilization of NFVI PoP considering packet length traffic variations sent by agents). Different to active agents that can be seen as generic benchmark VNFs, monitors need to observe particular properties according to NFVI PoPs or VNF profiles.

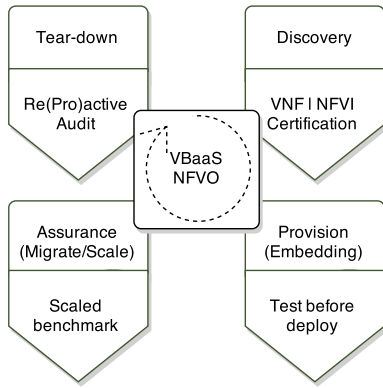


Fig. 5: VBaaS NFVO Processes

Manager: is mainly responsible for (i) the synchronization of activities between agents and monitors, (ii) collecting all benchmark raw results, and (iii) aggregating the inputs to construct a profile report that correlates different metrics as required by the VBaaS Profile. Therefore, it executes the main configuration, operation and management actions to deliver the results as specified by NFVI PoPs or VNF VBaaS profiles (e.g., instantiation of agents and monitors alongside tools and metrics configuration).

NFVI/VNF: defined as target by the orchestrator, which selects candidate PoPs or VNFs to be evaluated for the VNF-FG embedding problem or infrastructure certification procedures, for instance. It can be centralized or distributed depending on IB VBaaS profiles, NFVI PoP targets and VNFs characteristics.

IV. USE CASES

We now introduce VBaaS processes (see Fig. 5) to showcase its concepts and the envisioned benefits.

Discovery: means compliance certification for NFVIs and VNFs. In the first case, when attaching/extending an infrastructure domain, NFVO can use VBaaS to define performance certification procedures based on VBaaS IB profiles as *inputs* and infrastructure providers NFVI as *targets*. In the second case, the goal is to validate VNFs performance metrics. While the underlying processes are similar, each use case has different actors, i.e., network infrastructure providers and VNF developers, respectively. Prior to any of the other VBaaS NFVO processes, discovery is independent and is not directly associated with VNF-FGs definitions or their instantiation.

Provision: is a central motivation of VBaaS, where NFVO uses it to extract infrastructure resources information from different PoPs to offer performance views of customers VNF-FGs before deployment. In essence, it allows tailor-made NFVI PoPs with different guarantees of scalable performance based on IB VBaaS Profiles, which represent the VNFs under evaluation. A detailed provision use case is presented in the following subsection.

Assurance: for service providers it addresses their needs to guarantee expected infrastructure resources as VBaaS IB Profiles. For customers, assurance means the ability to verify expected performance levels according to their VNFs/VNF-FGs deployed and subject to (future) varying workloads. On

the one hand, it serves to validate the required deployed state of VNF-FGs while offering a complementary view to any (third-party or standalone/specific) continuous monitoring strategy. On the other hand, provided improved capabilities for scaling different NFVI PoPs, future (what-if) scenarios can be *a priori* benchmarked before the actual VNF-FGs migration/scaling. To this end, reproducible production traffic (think pcap traces) is instantiated by the VBaaS API to profile already working/desired features of the existent VNF-FG. In the first case, expected primitives of performance and consistency (e.g., reachability and isolation properties) are evaluated. For example, a network controller would configure “packet-outs” with different header profiles to bypass a firewall/IDS as defined in the VNF-FG to benchmark its deployed settings. In the second case (an arguably more complex scenario), existing features (e.g., stateful settings) of already deployed VNF-FGs would feed VBaaS Profile components (e.g., Agents) to verify test-before-(scale/migrate) properties of (already deployed) VNFs, with minimum performance requirements in candidate (targets) NFVI PoPs. Both processes act as a two-side benefit for customers and service providers to maintain KPIs while improving service agility and overall resource efficiency.

Tear-down: can be assisted with a VBaaS process to verify the cleaning steps of the remaining state of configurations left by previous VNF-FGs in NFVI PoPs, in proactive or reactive ways (think garbage collection). Firstly, when unsuccessful or unexpected end of a VNFs, VBaaS can verify the configured leftovers and serves as an incident analysis procedure to report possible malfunctioning/failed infrastructure or VNF settings. Secondly, tear-down can be used for security purposes (e.g., privacy maintenance of state configurations) or consistency checking (e.g., avoiding memory leaks and cleaning hardware/software specific profiles for new VNF-FGs).

A. VBaaS Provision Use Case

To illustrate the VBaaS provision process we now present an example of an orchestration process assisted by VBaaS. A generic description of VBaaS profile is used and all steps follow the activities in Fig. 6.

- 1) Upon receiving a customer network service request, the NFVO looks for required components as VBaaS profiles in IB.
- 2) NFVO creates a VBaaS VNF-FG with VNFs containing benchmark Agents, Monitors and Managers, and selects different candidate NFVI PoPs
- 3) VIM(s), receiving VBaaS VNF-FGs service graphs, perform the instantiation of VBaaS VNFs.
- 4) VBaaS Manager synchronizes Agents probes and Monitors, which execute the infrastructure benchmarking procedures. After receiving the results from Agents and Monitors, it aggregates, correlates, and finally sends the outputs to the NFVO via the VIM(s). For example, Agents probe bandwidth in the target PoP verifying packet loss and latency, while Monitors listen to PoP CPU and memory usage. Manager captures the results and aggregates them, generating VBaaS analytic results in addition to raw and associated metrics (e.g. latency vs. CPU).
- 5) VIM aggregates the results from possible different Managers and sends the upfront to the NFVO before

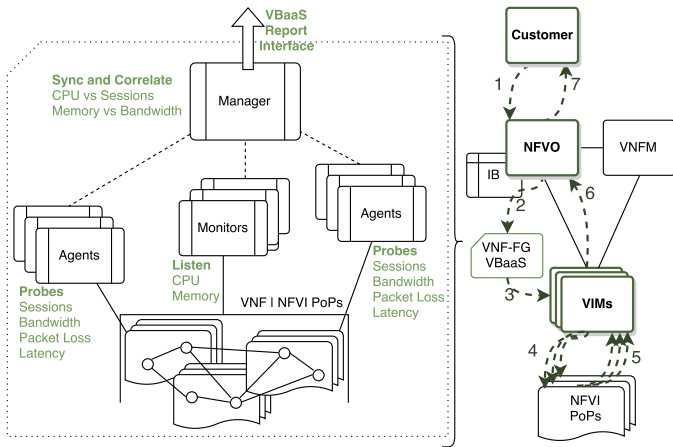


Fig. 6: VBaaS: VNF-NFVI Use Case

tearing down the VBaaS VNF-FG.

- 6) NFVO receives VBaaS reports and associates them with available infrastructure resources, informing customers of tailor-made NFVI PoPs candidates about the requested network service.
- 7) Customers (decision makers) analyse VBaaS reports and select best cost-value NFVI PoP to deploy their VNF-FGs with predictable performance.

V. VBaaS ANALYSIS

As VNF-FGs may appear with heterogeneous flavours and other NFV requirements (e.g., scale and migration capabilities), the actual performance is likely to be affected by available NFVI PoPs resources (cf. ETSI VNFs architecture recommendations [8]). Considering possible VBaaS strategies to extract reliable and consistent reports, as identified by [9], different types of evaluations can be performed considering:

When PoPs or VNFs need % of utilization: it is interesting to know about thresholds of resources usage to establish workload perimeters and capacity planning, for instance, to scale VNFs.

Noisy behaviour: when deploying VNF-FGs it is important to understand its behaviour in a shared infrastructure, causing any disturbance on neighbours, for instance, overhead or performance impacts.

Long-term tests: VNFs may require high reliability parameters (e.g., firewalls, IMS) and in this case will need extensible benchmarks to validate their long-term behaviour. Same can be applied to NFVI resources to supply VNFs requirements.

Constant flux: related with scale-(up/down) and (in/out) of VNFs and their infrastructure, benchmarks can be applied to verify the capacity of infrastructure agility in a constant flux to dynamically add/remove VNFs capabilities.

Recovery exam of failed VNF/NFVI: it can be considered in different approaches to provide resilience tests to obtain recovery metrics for VNFs and their respective infrastructure.

Shared resources can have particular cases to be considered in benchmarks. VBaaS needs to have prior knowledge about dependencies or interactions within NFVI PoPs resources.

Otherwise results would contain unexpected dependencies. It is an intrinsic NFVO task to recognize VBaaS profiles and infrastructure dependencies prior to perform any type of benchmark process. And the same must be considered when receiving and analysing VBaaS results, specially when requiring scaling of VNF-FGs.

On a related matter, benchmark comparisons between virtual and physical network functions profiles may be necessary, especially in levels where the first achieves comparable performance to the second, and how it can be scaled if there are capacity limits. In cases of black-box benchmarks (i.e., considering just external measurements) not always demanded resources are being used by VNFs, as demonstrated by [5]. Given the possibility to benchmark open VNFs, where internal parameters can be observed, integrating internal and external information can produce better analysis, specially for fine-tuning VNFs or their execution environment.

Metrics dependence needs important considerations, as there are different hardware/software domains that can be benchmarked based on a set of virtual layers to be instrumented and per-layer configurations. NFVO when initiating VBaaS processes needs to consider per virtual layer benchmark granularities and the set of different types of metrics to be evaluated on them. In addition, sampling rates define the possibility to create different metrics, as timely statistical analysis can be performed by NFVO over VBaaS results.

NFVO analysis requires previous knowledge about dependencies and interactions between multiple VNFs and their respective shared infrastructure, which can even be distributed. Therefore, VBaaS stability must be evaluated as depending on its targets, via calibration or tuning of VBaaS VNFs. It is possible that VBaaS VNFs suffer from workload variations jeopardizing the benchmark results. Besides VNF-FGs interconnections need to be carefully analysed in VBaaS, as VNFs, when chained, can behave differently or compromise others performance.

Trending results of VBaaS are shown in Fig. 7. According to VNFs and NFVI VBaaS profiles, depending on NFVI PoP, VNFs can behave differently according to their respective profiles. Also, VBaaS performed in different PoPs and time scales can present different results which would fit or not certain VNF profiles. It is an intrinsic VBaaS goal define the VNF profiles presented on the right side of Fig. 7. In the same way, it is fundamental to represent NFVI PoPs capabilities via NFVI profiles definition with equivalent representations of performance for different VBaaS profiles, as shown on the left side of Fig. 7. These profiles sustain the main motivations of VBaaS, which fill NFVI PoPs tailor-made capabilities for NFVO decision processes (e.g., embedding algorithms, capacity planning), aggregating value to customers preferences for their VNF-FGs allocation and certification.

For example, on the right side of Fig. 7 *VNF A* presents a different behaviour of CPU consumption associated to its bandwidth capacity when in comparison with *VNF B*, which seems to saturate CPU faster when its requirements of bandwidth increase. On the left side of Fig. 7, considering a generic VBaaS profile, we see it can present varying behaviours in different PoPs related to CPU and bandwidth equivalent consumptions. For example, *PoP C* shows linear increase of

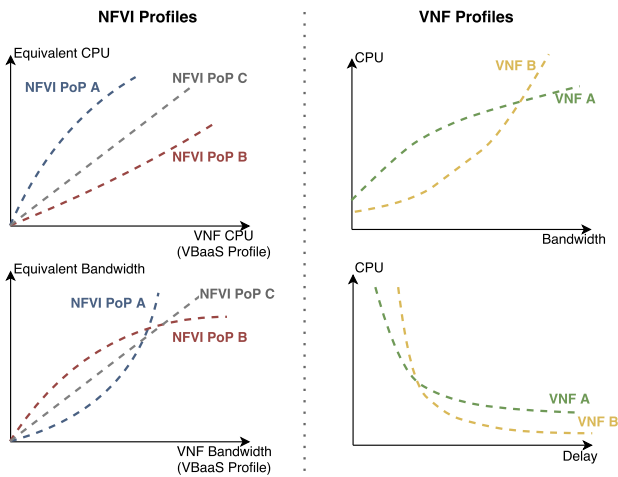


Fig. 7: Profiles Results

equivalent capabilities for VNF demands. Generally, among the examples illustrated, VBaaS results pursue the possibility to estimate and compare the predictable behaviour of VNFs in different PoPs.

Finally, VBaaS inherits, by design and implementation, the challenges of building a general framework to express methodologies of benchmarking for VNFs and NFVI PoPs. We proposed concepts and use cases to show initial requirements and considerations of VBaaS. Next steps include demonstrating a real case of VBaaS applicability on the extraction of a VNF profile to present a deep analysis about its associated cost, trade-offs related to its accuracy versus generality, and specially the importance of the presented requirements and considerations in this paper on the creation of performance profiles. In this way, we will provide insights on how VBaaS providers and consumers can understand when it should be used, how it could help continuous monitoring overhead savings, where it could certify deployed settings, and for which VNFs it would be possible benchmark and express the state of configuration dependent on infrastructure capabilities for estimating scenarios with predictable performance.

VI. CONCLUSION AND FUTURE WORK

This paper makes the case of VBaaS processes to provide: certification of VNFs and NFVI PoPs as performance profiles; and instrumentation prior to VNF-FG deployment as part of fine-tuning choices of VBaaS results with tailor-made PoPs.

VBaaS processes can be defined as simple tasks in common workflows of collection, synchronization, integration and export for benchmark analysis. Places where these tasks can take place would be defined according to VBaaS profiles, orchestrator policies and infrastructure capabilities. Based on VBaaS usage in NFVO processes, highlights in presented use cases contain interesting motivations for future and on-going work research opportunities, described below.

NFVO decision process for NFVI PoPs: the orchestrator must select infrastructure candidates as targets for VBaaS defining adequate parameters (e.g., time scale, metrics dependencies) for proper analysis of VBaaS and its VNF components.

Benchmark dependencies: VIM is responsible to define a series of consistent tasks for controllers perform the instantiation of benchmark components and procedures, which must consider their inter-operation dependencies consistently (e.g., deployment of VBaaS managers, agents and monitors).

Correlation of VBaaS results: NFVO is responsible to receive and correlate benchmark results following required VBaaS profiles aggregation levels and decision making process to build VBaaS reports consistently, attending customers needs.

Multi-domain SDN: Future work includes integration options with SDN architectures [10] as well as pursuing VBaaS-based inter-domain peering relationships [11].

Prototyping: Ongoing efforts are already devoted to map and evaluate VBaaS in the Unify [12] architecture.

ACKNOWLEDGMENT

This work was supported by the Innovation Center, Ericsson, Brazil, and in parts conducted within the framework of the FP7 UNIFY project, which is partially funded by the Commission of the European Union.

REFERENCES

- [1] R. Szabo, M. Kind, F.-J. Westphal, H. Woesner, D. Jocha, and A. Csaszar, "Elastic network functions: opportunities and challenges," *Network, IEEE*, vol. 29, no. 3, pp. 15–21, May 2015.
- [2] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu, "Research directions in network service chaining," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, Nov 2013, pp. 1–7.
- [3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *Communications Magazine, IEEE*, vol. 53, no. 2, pp. 90–97, Feb 2015.
- [4] ETSI GS NFV, "Architectural Framework," 12 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf
- [5] P. Veitch, M. McGrath, and V. Bayon, "An instrumentation and analytics framework for optimal and robust nfv deployment," *Communications Magazine, IEEE*, vol. 53, no. 2, pp. 126–133, Feb 2015.
- [6] E. Paik, M.-K. Shin, S. Pack, and S. Lee. (2015) Resource management in service chaining. Internet draft. [Online]. Available: <http://tools.ietf.org/html/draft-lee-nfvrg-resource-management-service-chain-01>
- [7] OPNFV. (2015) OPNFV Project Proposals. [Online]. Available: https://wiki.opnfv.org/project_proposals
- [8] ETSI GS NFV, "Virtual Network Functions Architecture," 12 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_NFV-SWA001v010101p.pdf
- [9] A. Morton. (2015) Considerations for benchmarking virtual network functions and their infrastructure. Internet draft. [Online]. Available: <http://tools.ietf.org/html/draft-morton-bmwg-virtual-net-03>
- [10] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
- [11] R. Rosa, M. Silva Santos, and C. Esteve Rothenberg, "MD2-NFV: The case for Multi-Domain Distributed Network Functions Virtualization," in *Networked Systems (NetSys), 2015 International Conference and Workshops on*, March 2015, pp. 1–5.
- [12] Unify. (2015) Unify Project Results. [Online]. Available: <http://www.fp7-unify.eu/index.php/results.html>