Using a Flexibility Measure for Network Design Space Analysis of SDN and NFV

Wolfgang Kellerer, Arsany Basta, Andreas Blenk
Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich, Germany
Email: wolfgang.kellerer, arsany.basta, andreas.blenk@tum.de

Abstract—**Flexibility is often claimed as a competitive advantage when proposing new network designs. However, most proposals provide only qualitative arguments for their improved support of flexibility. Quantitative arguments can quite vary among different proposals. A general understanding of flexibility is not yet clearly defined, leaving it to the reader to draw the right conclusions based on background information. The term flexibility is commonly defined as the ability to adapt to changes. For networks, flexibility would refer to the ability to adapt the available network resources, such as flows or topology, to changes of design requirements, e.g., shorter latency budgets or different traffic distributions. Recent concepts such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) have emerged to provide more flexibility in networks. Nevertheless, a deeper understanding of what flexibility means and how it could be quantified to compare different network designs remains open. In this paper, we propose a flexibility measure for network design space analysis and show its application. As it is quite challenging to formulate a flexibility measure that covers all network characteristics, we propose an initial set of flexibility aspects targeted to SDN and NFV. Moreover, we present the results of a detailed analysis of network function placement following either an SDN-based or an NFV-based approach. Our study reveals that in a logically centralized deployment scenario a mix of SDN and NFV provides the highest flexibility, with respect to our selected flexibility aspect.**

I. INTRODUCTION

Flexibility has become a key design objective for networks and respective proposed control and data plane mechanisms today. In fact, heterogeneous requirements from different application domains impose a high demand for networks to be designed for flexibility. These requirements include the ability to add new flows or even virtual networks on demand without influencing existing flows or networks, and the ability to temporarily extend a network topology to serve events, to give some examples.

Flexibility can be defined in different domains and from different viewpoints. For networks, which is the focus of this paper, flexibility refers to the ability of a network to adapt its resources such as flows or its topology to changes of requirements. This adaptation to changes may include the adaptation of the network configuration, the network topology, or the network functions and their placement.

In recent years, a number of technologies and approaches have emerged, which claim to provide flexibility in networks. One widely accepted approach is the concept of Software Defined Networking (SDN) [1] separating the data plane from a logically centralized control plane with a standardized interface allowing programmability and hence providing flexibility to networks. SDN-based network control can be complemented by the concept of Network Virtualization (NV) [2] where network resources can be operated on logical, hence, virtual level on a physical network substrate. With NV, virtual networks may change over time in order to adapt to time-varying requirements from different application domains [3]. The concept of virtualization has also been extended to network functions. Network Function Virtualization (NFV) [4] aims at providing network functions such as gateways and middleboxes in software, which can be run on commodity hardware, e.g., in data centers.

SDN and NFV, which are in the focus of this paper, target different aspects of flexibility. SDN was developed to target programmable flows and to centralize network control, which contributes to flexibility in terms of flow steering and configuration. This flexibility can be assessed in terms of the number of possible configurations. For instance, OpenFlow (OF) [1], which is the most commonly used protocol to implement SDN, has an upper boundary in its flexibility due to the limited set of configurations defined in the specification of each OF protocol version. Flow configuration based on application layer information, e.g., HTTP or video URLs, from flow packets are not yet available with the latest OF protocol version 1.5 [5].

NFV leverages virtualization to functionality, where functions get developed as software and are executed on commodity hardware. Having programmable hardware can offer flexibility to define and program function operation including the configuration of functions, the extension of functionality and the change of whole functions. NFV can also provide flexibility in terms of function scaling by being independent from networking hardware, e.g., scale up resources assigned to a network function or scale out a function on multiple hardware entities. Software functions, which are independent from hardware, contribute to the function placement flexibility [6].

Although SDN and NFV evolved as new technologies that increase the ability of a network to be adapted, a clear definition of what flexibility means for networks is still missing. Moreover, there is no common agreement on a quality measure quantifying a network's flexibility. Such quality measure could be defined similar to what has been defined for Quality of Service (QoS). QoS has been introduced to provide a common understanding about network support for service level performance aspects, e.g., data rate, delay and jitter.

In this paper, we advocate the necessity to come up with a network flexibility measure consisting of a common set of flexibility aspects. Similar to QoS, where the importance of aspects such as data rate and delay varies among different service requirements, flexibility depends on the requirements as well. For instance, the placement of functions may be important for some network designs, for others it is is the scale in topology size. Hence, we are not aiming at quantifying flexibility of networks as a singular comparative measure, but rather through a set of flexibility aspects. To be able to quantitatively compare different network designs with respect to their flexibility, a common definition of main flexibility aspects is indispensable.

The main contributions of this paper are

- an analysis of the state of the art with respect to the use of flexibility as a measure,

- a proposal for an initial set of flexibility aspects for a common network flexibility measure targeted to software-driven networking based on SDN and NFV, and

- a detailed design space analysis of a function placement scenario based on a selected flexibility aspect.

In the remainder of this paper, we analyze state of the art approaches for their support of flexibility in Section II. In Section III, we propose a selection of flexibility aspects for a network flexibility measure targeting SDN and NFV. Section IV demonstrates the usage of a quantitative measure for flexibility with an analysis of a function placement scenario in a mobile core network.

II. FLEXIBILITY IN THE STATE OF THE ART

In this section, we analyze the state of the art that argues about flexibility in general, i.e., related work that does not specifically target SDN or NFV. We extract the definition of flexibility applied in each targeted use case and, if provided, show how flexibility is expressed via quantitative parameters. Note that an extended related work is presented in [7].

A. Flexible Traffic Control

Vissicchio et al. [8], [9] introduce Fibbing, an architecture that "readily supports flexible load balancing, traffic engineering, and backup routes". Fibbing provides a way to have a control plane that runs physically distributed but is still centrally controlled. For this purpose, they introduce fake nodes and links in order to indirectly impact the path calculation of the distributed control plane. Hence, the advantages of both worlds , i.e., physically distributed deployment and logically centralized control, should be combined. The authors also mention that "while more flexible (e.g., enabling stateful control logic) than Fibbing, SDN requires updating the switch-level rules one-by-one" thus "forgoes the scalability and reliability benefits of distributed routing."

B. Flexible Network Architectures

Anderson et al. [10] discuss the flexibility gain of network virtualization. They argue that virtualization is needed in order to provide flexible experimentation with traffic from the current Internet. Furthermore, they introduce two views of a future architecture, the purist view and the pluralist view. As the architecture remains in place a long time, the purists aim for architectural flexibility. This means that the architecture should only provide mechanisms to be changed over large time scales. In contrast, the pluralists want to provide the ability to add or augment overlay networks when needed. They argue that flexible adding or removing overlays, i.e., changing virtual topologies, provides the needed flexibility.

The authors of [11] investigate the flexibility of inserting new technologies in existing architectures. They define "Flexibility" as "the ability for" an "approach to adapt to changes in topology over time (...) as well as failures". They quantify the flexibility for different technologies. For instance, flexibility (fault tolerance) vs. achieved throughput. While one approach is more flexible (failure resilient), it adds overhead, thus, decreases throughput.

C. Flexible Mobile Networks

Jin et al [12] tackle the challenges of the cellular core network. They say that the current mobile core network is "inflexible" for three reasons: they "forward all traffic through the P-GWs", "P-GWs are not modular", carriers cannot "mix and match capabilities from different vendors (e.g., use a firewall from one vendor, and a transcoder from another)". They propose the scalable architecture SoftCell that can make fine-grained policies for the mobile core network devices. SoftCell uses so called flexible, high-level service policies. Operators can use these policies to redirect traffic through middleboxes, which are operated according to the demands of subscribers. The high level policies are realized via switches that are deployed close to the base stations. The core switches enable forwarding to the needed middleboxes, i.e., network functions.

D. Flexible Network Management

Arumaithurai et al. [13] propose Function-Centric Service Chaining (FCSC). FCSC is based on Information Centric Networking in order to make the management of networks that use virtualization for dynamic function placement more flexible. They see flexibility as the ability to adapt faster to failures and to change middleboxes more quickly. More in detail, "an efficient service chaining network should support (...) changes in a flexible way - (...) middleboxes should be able to determine the functions of a flow themselves and the changes should take effect immediately."

E. Flexible Data Plane

Bosshart et al [14] propose to make the current SDN switch hardware more flexible and "that flexible OpenFlow hardware switch implementations are feasible at almost no additional cost or power." In general, they say that "Flexible processing is achievable via many mechanisms. Software running on a processor is a common choice." Using their Reconfigurable Match Tables, a network designer should be equipped to change how a switch processes network headers at runtime. In detail, the forwarding plane is modified without changing the hardware.

Similar to the previous concept, Hwang et al. [15] say that software solutions running on commodity servers, whose hardware is extensively exploited via software extensions, e.g. DPDK, provides "far greater flexibility" than existing purpose-built hardware. They propose NetVM, which "enable(s) in-network services", e.g., firewalls or proxies, "to be flexible created, chained and load balanced."

In summary, we can observe that flexibility is mentioned as a key argument when proposing new network designs. However, several different perspectives of flexibility are considered, while a common understanding of network flexibility as a measure is missing so far. Nevertheless, some related work has aspects of network flexibility in common, such as adapting the network resources according to changing demands over time. Based on our observations, we are going to propose a set of flexibility aspects as part of an initial common measure for the concepts of SDN and NFV.

III. PROPOSING FLEXIBILITY ASPECTS

At the moment, there is no unified measure that can express how flexible a network is, i.e., to quantify flexibility for comparing network design choices for SDN and NFV. In general, network flexibility as a measure is mostly used with a specific objective in mind, e.g. "network A can be adapted faster to changing requirements than network B", focusing on some selected set of parameters, which we could already observe in Section II. Hence, in order to come up with a common measure, the challenge is to find reasonably independent flexibility aspects combining some of the parameters to support an intuitive understanding of flexibility.

A network can be assessed in terms of its flexibility to adapt its task, i.e., its functionality, to a diverse set of demands, e.g., different data plane latency needed over time. In detail, to change the functionality of a network, the network resources and how they are used, i.e., operated, has to be changed accordingly. This involves resources of the network nodes and the network links. Both, SDN and NFV nowadays provide a more flexible way to adapt the use of those resources. For instance, SDN provides a possibility to change the routing strategy of a network at run time, i.e., it provides a flexible **flow steering** and configuration. Besides, the functionality of a network is assembled of its individual network functions and their **placement** inside the network. A flexible network function placement allows, e.g., to meet different latency requirements and also the combination of functions, e.g., a diverse chaining of functions. Furthermore, the ability of the functions to be flexibly **scaled** and **operated** is another essential feature for flexibility. Flexible scaling, for instance, can meet demands as such of instant flash crowds. A flexible operation meets the demands of requirements for different networking use cases, e.g., different applications like video streaming or voice over IP (VoIP).

Note that the proposed aspects can be considered as lenient examples for an initial set of network design choices in the context of flexibility of SDN and NFV. These flexibility aspects can be extended further through new networking concepts, technologies, or future design requirements. A more detailed discussion of flexibility aspects can be found in [7].

A. Flow Steering

Flow steering and configuration describe the course of flows inside a network through configuring a forwarding policy for a flow on each network hop. Flow steering can be considered as an elementary attribute for flexibility. Flexibility can be related to the magnitude and granularity of flow steering alternatives. More possible steering strategies would reflect to higher flexibility. As an example, a network element that can support only forwarding of packets is less flexible than an element that can provide both forwarding and duplicating packets on multiple ports for instance. Flexibility of flow steering can be also coupled with the time required to change. Network elements can vary from not being able to change the flow steering on run-time, i.e., static, to elements that can support run-time flow steering. It is important to note that there are possible side-effects as a result of higher flexibility in terms of flow steering. From an operational aspect, changing the steering always requires additional control, which imposes latency and data overhead.

B. Function Placement

A flexible function placement has a direct impact on the network performance, e.g., the SDN controller placement with respect to switches and its impact on control latency. Dynamic placement adds an additional dimension to flexibility in case changing the function placement is supported through, e.g., migration techniques for virtual functions. The placement flexibility is directly influenced by the set of possible locations to place a function. More potential locations have the degree of freedom to place network functions such that diverse or even more strict requirements, e.g., changing latency requirements, can be satisfied. A dynamic function placement that can change on run-time offers more flexibility than a static placement.

C. Function Scaling

Supporting function scaling means that the assignment to network functions can be adapted at runtime. It is decided based on the possible resources, e.g., network element CPU or link capacity, that can be allocated to individual functions. For example, for a network element that has two functions which share equally its resources, e.g., CPU or memory, resource allocation flexibility would mean that we can assign 80% of the resources to one of the functions. High function scalability would be achieved with support of frequent changes of resource assignments between the functions. Flexibility is also related to the granularity of such resource assignment. Note that adding more resource allocation flexibility in network elements means more complexity and management overhead.

D. Function Operation

Function operation denotes the ability of configuring the functionality of network elements. This means that the operation of a network element can be changed between, e.g., firewalls, NATs, load balancers, etc. Nowadays, programmable switches are being introduced which allow the operator to change and tweak their network function. Hence, a programmable configurable network element can be another driver to increase network flexibility. Flexibility of function operation can be assessed in terms of the set of possible functions

supported by the programmable network element. The run-time support to change the function operation can be considered as another main enabler for higher flexibility. The cost of flexible function configuration can be observed in terms of latency or control overhead. A flexible function operation might also impact the performance of the data plane. For instance, if the function operation is only supported by the software part that runs on, e.g., general purpose switches, i.e., not in hardware, a drastic decline in performance might be observed compared to operations supported by the switch hardware.

IV. Analysis for Function Placement Aspect

In this section, we demonstrate an initial example for one quantitative measure for flexibility with an analysis for the function placement aspect comparing SDN and NFV network design choices. This example considers the application of SDN and NFV concepts to the mobile core network. SDN and NFV promise to bring more flexibility to the core network, however, a quantitative analysis to such flexibility as we provide in the following has not been reported in the state of the art so far. Note that the flexibility analysis for the function placement aspect is intended to demonstrate the viability of our proposal. The analysis would be extended with other flexibility aspects in future work.

A. Design Choices

A typical mobile core architecture is depicted in Fig. 1a, where we focus on core network functions that handle the flow setup control and data plane forwarding, i.e., Serving Gateway (SGW) and Packet Data Network Gateway (PGW). There are three design choices for a next generation mobile core network under investigation [16]. The first design choice is a full SDN deployment. The mobile core gateways are decoupled into data plane and control plane parts, i.e., SDN switches and SDN controllers, respectively. The controllers are centralized and deployed on available cloud computing platforms, i.e., data centers (DCs), while switches can be distributed along the network, as shown in Fig. 1b. This has an impact on the control plane performance, e.g., latency, since the core network control plane is additionally extended by SDN control.

The second design choice is a full NFV deployment, illustrated in Fig. 1c, where the gateway functions run as virtual software functions deployed at data centers. This implies that both data and control planes are operated at the data centers, hence, latency depends on the location of the data centers.

Fig. 1d shows the third design choice, which is a mixed SDN and NFV deployment. The gateway functions are partially operated through SDN (design choice 1) while the other part is run as virtual software functions on data centers (design choice 2). This third design choice offers another degree of freedom to match the latency requirements for each mobile core network function individually.

B. Problem Setup

Moving towards a next generation mobile network [17] [18], data and control plane latencies are one of the most important performance metrics in a mobile network for both users and operators. Data and control latency performance
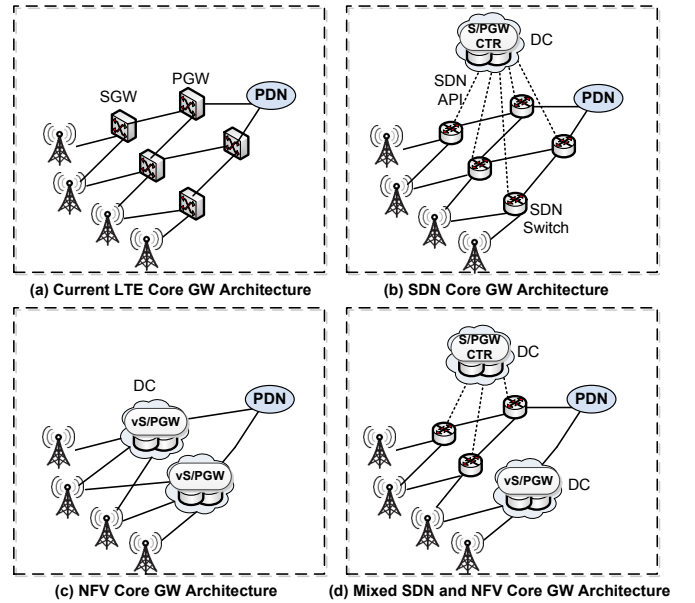

Fig. 1: Mobile Core Architectures based on SDN and NFV

is influenced by several parameters. First, the latency performance depends on the chosen design, i.e., SDN only, NFV only or a mixed SDN and NFV deployment. Second, the latency performance depends on the location of the data centers. For SDN the control latency is impacted, while for NFV the data center location impacts both data and control latencies. Additionally, the number of available data centers contributes to the latency performance by constraining the placement of the network functions for either SDN or NFV, e.g., a rather centralized vs. a distributed function placement.

In our previous work, we have introduced this function placement problem as a mixed integer linear program (MILP) formulation that models the different design choices and includes data latency constraints. The evaluated topology is a core network based on the US population distribution and the coverage of a tier 1 operator in the US, which consists of 18 SGWs and 4 PGWs. The mathematical formulation and the used topology can be found in [19]. The original formulation has been extended to model control traffic paths and latency constraints for the control plane of the three core network designs.

The target of this analysis is to investigate the flexibility of the three SDN and NFV core network designs in terms of their ability to adapt and cover a wider range of data and control latency requirements. In other words, we investigate how flexible a network design is, to change and support different requirements. This is evaluated by solving the function placement problem with several data and control latency requirements, from which the flexibility measure is derived. Note that the evaluation is objective agnostic as we are concerned if a network solution is possible to serve the changing latency constraints, regardless of the objective, i.e., cost. For future work, we consider evaluating the trade-offs between network flexibility and its cost.
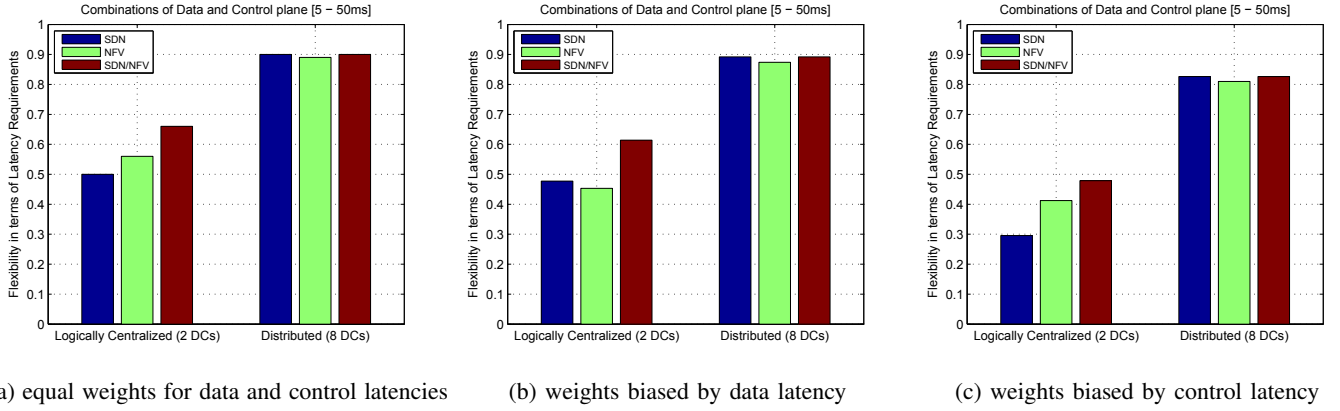
| (a) equal weights for data and control latencies | (b) weights biased by data latency | (c) weights biased by control latency |

Fig. 2: Flexibility measure comparing SDN and NFV design choices, for logically centralized and distributed DC infrastructure

C. Flexibility Measure and Parameters

The flexibility of the three design choices is evaluated with 10 latency requirements for each data and control, where data latency $lat^d = \{5, 10, ..., 45, 50\}$ ms and control latency $lat^c = \{5, 10, ..., 45, 50\}$ ms. This results in a total of 100 combinations of data and control latency requirements to be evaluated. The function placement problem is solved for each combination of data and control latency requirements. A binary variable $feasibleSol_{i,j}$ is defined that gets a value of 1 in case a function placement solution exists under the constraints of lat_i^d and lat_j^c, 0 otherwise.

In order to convey the significance or difficulty of finding a solution under a combination of data and latency requirements, $feasibleSol_{i,j}$ is weighted by a factor $w_{i,j}$. The purpose of the weights is to represent that a network design which supports latency requirements, e.g., down to 5 ms, would be more flexible compared to another design that only supports, e.g., down to 20 ms. The flexibility measure $Flex$ is given in Eq. 1, where the sum of weighted feasible solutions is normalized by the sum of all weights for a unified comparative representation.

$$Flex = \left(\sum_i \sum_j feasibleSol_{i,j} * w_{i,j}\right) / \left(\sum_i \sum_j w_{i,j}\right) \quad (1)$$

The weights $w_{i,j}$ are defined in Eq. 2. The constant α reflects the significance of data latency, while β represents the importance of the control latency. The weights consider the reciprocal of the latency values to provide higher values for lower latency requirements.

$$w_{i,j} = \frac{\alpha}{dataLatency_i} + \frac{\beta}{controlLatency_j} \quad (2)$$

Three different weight distributions are evaluated to cover different operator strategies to quantify the network flexibility. The first weight distribution represents the case where all data and control latency combinations have equal importance. This means that the flexibility measure quantifies the number of possible supported latency combinations, regardless of their significance, i.e., values. This is realized by setting all weights to be equal, i.e., $\forall w_{i,j} = 1$.

The second weight distribution defines higher weights for finding a solution with stricter data latency requirements. This represents the case where flexibility is required more for the data plane rather than for the control plane. It is reflected on the weight distribution by setting the value of $\alpha >> \beta$ in Eq. 2. For evaluation, we used $\alpha = 10$ and $\beta = 1$. Finally, the third weight distribution defines higher weights for finding a solution with stricter control latency requirements. This reflects the case where flexibility is required more for the control plane, e.g., with critical control plane services used in industrial automation. For this distribution, $\alpha = 1$ and $\beta = 10$ are used for evaluation.

D. Flexibility Observations

The flexibility measure based on the function placement aspect in this initial investigation is evaluated for two possible data center infrastructure scenarios. The first scenario is for the case an operator logically centralizes its data center infrastructure into 2 data centers compared to the second scenario with distributed data center infrastructure with 8 data centers. Note that in the evaluated topology, there are 22 possible locations to place the available number of data centers.

Initial evaluation results for the flexibility of the three design choices, namely SDN, NFV and mixed SDN/NFV, are shown in Fig. 2. Given a logically centralized data center infrastructure for all evaluated weight distributions, a mixed SDN/NFV design choice gives the highest flexibility compared to an SDN only or an NFV only design. This means that a mixed SDN/NFV design can support more data and control latency requirements as well as stricter latency budgets. That means further that a combined SDN/NFV design would show more flexibility in case a change is required for the latency requirements by the operator or the regulators.

Given a distributed data center infrastructure for all evaluated weight distributions, the observed flexibility in terms of covering latency requirements is quite similar for the three design choices. This implies that an operator can go with any of the three design choices and can have the same flexible support for the change of latency requirements. This

illustrates where the flexibility differences disappear between the different design choices, which might be an advantage for the operator to be able to choose any of the designs.

Comparing the three evaluated weight distributions for the flexibility measure, there are trade-offs between the SDN and the NFV design for a logically centralized data center infrastructure. With equal weights illustrated in Fig. 2a and control biased weight distribution shown in Fig. 2c, the NFV design choice shows more flexibility than the SDN design. With equal weights, the results show that NFV can support a higher number of data and control latency requirements than SDN. While with control biased weights, NFV shows more flexibility than SDN by supporting stricter control latency budgets, since SDN adds more control latency to the conventional core network control. On the other hand, regarding a data biased weight distribution illustrated in Fig. 2b, the SDN design choice shows more flexibility than the NFV design. That is due to the fact that SDN only consolidates and centralizes the control plane, while the data plane can be distributed to offer better data plane latency. NFV centralizes both data and control which might violate several of the stricter data latency requirements due to additional transport of data traffic to consolidated functions at data centers.

V. Conclusion

Flexibility is commonly used as a differentiating feature in recent proposals for network designs. However, quantitative arguments are often missing in order to express clearly which flexibility aspects are addressed to which extent and why is one network design more flexible than another. We claim that with emerging networking concepts such as SDN and NFV, network flexibility will most likely become one of a new measure in network research and development in the future. In our initial proposal such flexibility measure is not a single parameter but includes several flexibility aspects. Addressing the flexibility provided by SDN and NFV, in particular, these aspects include the ability to configure the forwarding policy for a flow, to place network functions, to scale the resources allocated to network functions and to configure and change network functions.

We demonstrate a quantitative measure for flexibility for the function placement aspect comparing three SDN and NFV design choices for a mobile core network. Flexibility of the different design choices is expressed in terms of their support for data and control latency requirements. Our observations show that for a centralized deployment scenario a mix of SDN and NFV provides the highest flexibility. For a distributed data center infrastructure used for deployment the observed flexibility is almost the same for the three design choices. Hence an operator could choose any one of them when focusing on flexibility as a design criteria for implementation. For future work, further aspects would be considered to quantify the network design flexibility. Additionally, we are interested to find the trade-offs between the flexibility of a network design and its induced cost.

Acknowledgment

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM CRR*, vol. 38, no. 2, p. 69, 2008.

[2] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, pp. 862–876, 2008.

[3] A. Blenk and W. Kellerer, "Traffic pattern based virtual network embedding," in *ACM CoNEXT Student Workshop*, SF, CA, USA, pp. 23–26, 2013.

[4] "ETSI GS NFV-MAN Network Functions Virtualization (NFV); Management and Orchestration v1.1.1," Dec. 2014, http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf.

[5] ONF, "OpenFlow Switch Specifications 1.5.0," Oct. 2012, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

[6] A. Basta, A. Blenk, M. Hoffmann, H. J. Morper, K. Hoffmann, and W. Kellerer, "SDN and NFV dynamic operation of LTE EPC gateways for time-varying traffic patterns," in *6th International Conference on Mobile Networks and Management (MONAMI)*, pp. 63–76, 2014.

[7] W. Kellerer, A. Basta, and A. Blenk, "Flexibility of networks: a new measure for network design space analysis?" *arXiv preprint arXiv:1512.03770*, 2015.

[8] S. Vissicchio, L. Vanbever, and J. Rexford, "Sweet Little Lies: Fake Topologies for Flexible Routing," in *ACM HotNets Workshop*, pp. 1–7, 2014.

[9] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford, "Central Control Over Distributed Routing," in *ACM SIGCOMM*, 2015.

[10] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[11] M. K. Mukerjee, D. Han, S. Seshan, and P. Steenkiste, "Understanding tradeoffs in incremental deployment of new network architectures," in *ACM CoNEXT*, pp. 271–282, 2013.

[12] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *ACM CoNEXT*, pp. 163–174, 2013.

[13] M. Arumaithurai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan, "Exploiting ICN for flexible management of software-defined networks," in *INC*, pp. 107–116, 2014.

[14] P. Bosshart, G. Gibb, H.-S. Kim, G. Varghese, N. McKeown, M. Izzard, F. Mujica, and M. Horowitz, "Forwarding metamorphosis," in *ACM SIGCOMM*, p. 99, 2013.

[15] J. Hwang, K. K. Ramakrishnan, and T. Wood, "NetVM: High Performance and Flexible Networking Using Virtualization on Commodity Platforms," in *USENIX NSDI*, pp. 445–458, 2014.

[16] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "A virtual SDN-enabled LTE EPC architecture: a case study for S-/P-gateways functions," in *IEEE SDN for Future Networks and Services (SDN4FNS)*, pp. 1–7, 2013.

[17] 5G Initiative Team, "NGMN 5G Initiative White Paper," Feb. 2015, https://www.ngmn.org/uploads/media/NGMN-5G-White-Paper-V1-0.pdf.

[18] Mobile and wireless communications Enablers for the Twenty-twenty Information Society (METIS), "Final report on architecturer (Deliverable D6.4)," Feb. 2015, https://www.metis2020.com/wp-content/uploads/deliverables/METIS-D6.4-v2.pdf.

[19] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways; the functions placement problem," in *4th Workshop on All things Cellular: Operations, Applications, & Challenges*. ACM, pp. 33–38, 2014.