ACCEPTED FROM OPEN CALL

Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges

Ahmed M. Medhat, Tarik Taleb, Asma Elmangoush, Giuseppe A. Carella, Stefan Covaci, and Thomas Magedanz

The authors introduce a service function chaining taxonomy that considers architecture and performance dimensions as the basis for the subsequent state-of-the-art analysis. The article concludes with a gap analysis of existing solutions and the identification of future research challenges.

ABSTRACT

Service function chaining is a network capability that provides support for application-driven-networking through the ordered interconnection of service functions. The lifecycle management of service function chains is enabled by two recently emerged technologies, software defined networking and network function virtualization, that promise a number of efficiency, effectiveness, and flexibility gains. This article introduces a service function chaining taxonomy that considers architecture and performance dimensions as the basis for the subsequent state-of-the-art analysis. The article concludes with a gap analysis of existing solutions and the identification of future research challenges.

INTRODUCTION

Network resource management and service differentiation according to user requirements and network constraints are crucial elements of the business and operations support systems of any telecommunications operator. These two key capabilities are particularly challenging considering the steady increase in the number of services/applications, their heterogeneous quality of service (QoS) requirements, and the overall traffic that the network has to provide. Service function chaining (SFC) is an enabling technology for the flexible management of specific service/application traffic, providing solutions for classifying flows and enforcing adequate policies along the flow routes according to the service requirements and considering the availability status of the network. SFC is defined as a chain-ordered set of service functions (SFs) that handles the traffic of the delivery (data plane), control, and monitoring (control plane) of a specific service/application.

Recently, SFC has made use of the new technology called software defined networking (SDN). Architecturally seen, SDN decouples the control plane from the data plane and introduces appropriate programming abstractions exploited in SFC for the dynamic control of the topology of SFCs and the traffic steering across SFs. Network function virtualization (NFV) is related to the telco initiative of adopting cloud-computing technology enabling the virtualization of software-implemented network functions (SFs in SFC terminology). NFV is adopted by SFC to provide efficient and effective deployment and orchestration of SFs.

The SFC architecture specifications are addressed by the IETF SFC working group (RFC 7665) and the Open Network Foundation (ONF). SFC becomes particularly relevant in the new emerging value chains involving multiple data centers (central, edge, fog), access-, core- and transit-networks, and application service providers. As such, SFC has attracted much attention within the community of researchers as well as among network operators and network equipment vendors (e.g., Juniper, QOSMOS, and Huawei). Numerous open source tools enabling SFC are also available. Notable examples are OpenDaylight, OPNFV, ONOS, OpenContrail, and OpenStack's Neutron/Service Insertion and Chaining.

The main contributions of this article are twofold. First, the article explores the limitations of current SFC approaches in next generation networks in terms of architectural and conceptual research work by providing a brief analysis of each solution in the state of the art. The limitations are explored with reference to the SFC IETF specification. Second, the article draws some new research directions. To the best knowledge of the authors, this article is the second research work highlighting the limitations of SFC approaches and the first work to provide a detailed overview of the SFC state of the art and evaluation. The work introduced in [1] was the first in defining the new research directions and challenges of SFC. The authors in [1] presented SFC design considerations and requirements with use cases that show the advantages of adopting SFC. Their main contribution is to explore the research challenges during the exemplary lifecycle of an SFC in an applicable telco network, covering SFC definition, deployment, programming, and security concerns.

The rest of the article is organized as follows fashion. We illustrate the SFC standardized architectures, as defined by the IETF SFC and ONF working groups, and discuss how the ETSI NFV architecture provides SFC. We highlight previous research work conducted on the SFC architectural concepts and implementations. SFC challenges and limitations are discussed. Finally, the article concludes.

Digital Object Identifier:
10.1109/MCOM.2016.1600219RP

Ahmed M. Medhat, Asma Elmangoush, Giuseppe A. Carella, and Stefan Covaci are with Technical University Berlin; Tarik Taleb is with Sejong University and Aalto University; Thomas Magedanz is with Fraunhofer FOKUS

0163-6804/17/$25.00 © 2017 IEEE

SFC Standardized Architectures

According to the IETF SFC specifications (draft-ietf-sfc-control-plane-06), a typical SDN-based SFC architecture consists of components grouped into two layers, the control plane and data plane, as shown in Fig. 1. The control plane is responsible for the SFC management, SF instances management, mapping SFC to a specific service function path (SFP), installing and administering forwarding rules on the service function forwarding (SFF) components of the data plane, and adjusting the SFP in terms of SF instances and overlay links as a result of their status (i.e., overloaded, active, inactive, failed, etc.). The SFC control plane components interact with the SFC data plane components via four interfaces. The first interface C1 is responsible for pushing the SFC classification rules defined by the SFC control plane into the SFC classifiers. The SFFs report the connectivity status of their attached SFs to the SFC control plane. Interface C3 is between the NSH-aware SFs and the SFC control plane. It is used to collect some packet-processing statistics (e.g., SFs' load update) from the SFs. For NSH-unaware SFs, a SFC proxy is provided for collecting statistics (e.g., SF processing latency and workload) and transmitting this information over the C4 interface to the SFC control plane. The SFC control plane uses these statistics (received through interfaces C2, C3, and C4) to dynamically adjust the SFPs.

The main components of the SFC data plane, as shown in Fig. 1, are the SFC classifier, SFF, SF, and SFC proxy. The SFC classifier differentiates the incoming traffic into flows, based on the target application and other predefined requirements. The SFC classifier tags each flow by adding an SFC header containing a service function path (SFP) ID to each flow packet header. The path ID is related to an SFC and identifies the ordered set of abstract SFs which must be performed to the particular flow. The SFP is the real path (the exact SFFs/SFs) that packets traverse.

An SF executes a particular set of actions on incoming packets (e.g., deep packet inspection or firewall functions) and can process packets belonging to several SFPs. An SF can be present with multiple, distributed instances in the network (e.g., for scalability reasons). An SFF is in charge of sending the incoming traffic to SFs and/or other SFFs, according to the defined SFPs. To this purpose, the SFF uses and inserts SFP-specific information in an additional packet-header (SFP packet encapsulation). The IETF SFC working group does not standardize a particular SFF, but instead the SFC special header, called the network service header (NSH) (draft-ietf-sfc-nsh-02). An SFC proxy may become required between SFF and SFs as the majority of SFs do not recognize the SFC packet headers (NSH). The SFC proxy performs SFC packet de-capsulation for the packets forwarded to the NSH-unaware SFs and encapsulates these packets before sending them to the SFF (IETF SFC RFC 7665).

In the SDN context, the Open Networking Foundation (ONF) also proposed another model for the L4-L7 SFC architecture, based on the SDN/OpenFlow controller (ONF TS-027). The ONF SFC system is based on the IETF SFC specification in that it specifies the SFF by an extended OpenFlow switch version supporting NSH.

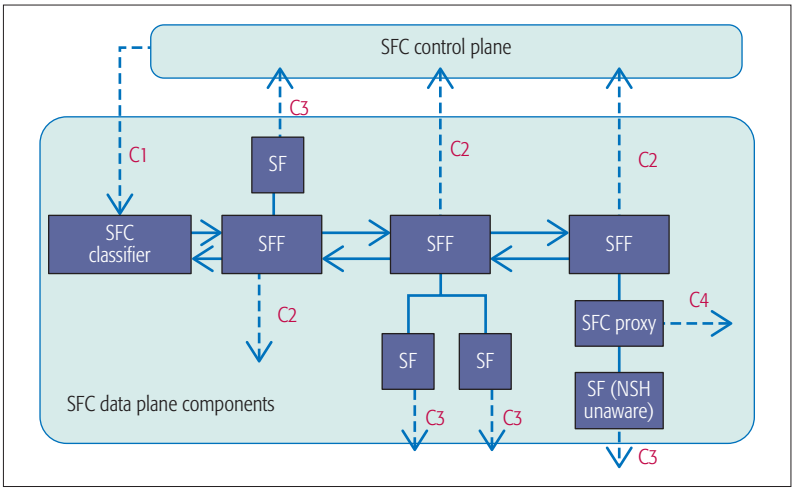

Figure 1. A typical SFC architecture (draft-ietf-sfc-control-plane-06).

An SFC control plane functional architecture is addressed by the ETSI NFV architecture (ETSI GS NFV-MAN 001 V1.1.1) (see Fig. 2). The main components of the ETSI NFV architecture are: NFV orchestrator (NFVO), virtual network function manager (VNFM), and virtualized infrastructure manager (VIM).

NFVO is responsible for the end-to-end management and orchestration of network services (NS) provided by an administrative domain. Each NS is specified by a network service descriptor (NSD). An NS may span multiple network domains belonging to the same or server different administrations. Each network domain contains a network level manager called the network controller that is responsible for network connectivity management. In the case of an NS spanning multiple administrative domains, the overall end-to-end management of the NSs is realized by co-operation of the participating NFVOs, either in a hierarchical or in a peer-to-peer manner. In the case of the hierarchical arrangement, an additional NFVO is introduced in the architecture. Each virtualized infrastructure domain is managed by the so called VIM (e.g., in the case of OpenStack, the virtual network infrastructure manager is the neutron component). NFVO is also concerned with instantiating/updating/terminating of SFCs (i.e., life cycle management of the SFC) and their constituent VNFs (instantiation, update, scaling, migration, and termination) in coordination with VNFMs. The VNFM is responsible for VNFs life cycle management such as VNFs instantiation, update/upgrade, scaling, and termination. The VIM is concerned with controlling and managing the NFV infrastructure (NFVI) compute, storage, and network resources such as providing a "Network as a Service" northbound interface to the higher layers (NFVO and VNFM) and invoking the NFVI network southbound interfaces (network controller or/and VNFs/PNFs) to construct the service within the domain. Each NS contains at least one VNF forwarding graph (VNFFG) that describes the network topology of the NS or a portion of the NS by referencing the VNFs, PNFs, network forwarding path (NFP) that provides the order of involved VNFs or PNFs in the VNFFG, and the virtual links that connect them. In SFC terminology, the VNFFG is considered as the SFC,

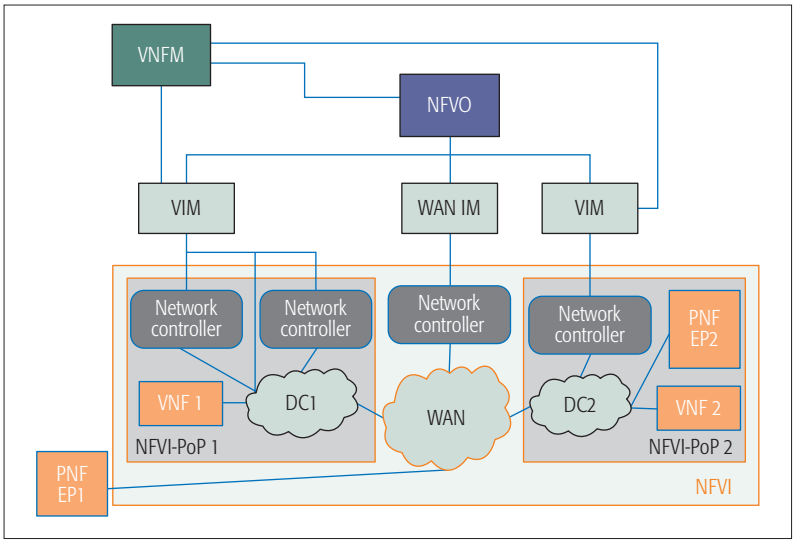

Figure 2. ETSI NFV Architecture (ETSI GS NFV-MAN 001 V1.1.1).

VNFs or PNFs are the SFs, NFPs are the SFPs, and Virtual Links are implemented by one or different SFFs. Fig. 3 shows an example of two VNFFGs (SFCs) imbedded in the same virtual network infrastructure.

Recent research works, such as the work presented in [2], exemplify how SDN/NFV-based SFC standardized solutions can be applied to solving severe congestion in mobile networks (access and core networks) caused by the exchanged user generated content of mobile social media applications through mobile devices.

State of the Art of SFC Concept and Implementations

A wide range of research work has been conducted proposing new frameworks, concepts, and implementations of SFC. These approaches can be classified into two categories based on the adopted technology (i.e., SDN and NFV). Different SFC solutions are investigated, compared, and evaluated in this section, discussing their limitations and defining the research directions that should be considered in the future to improve them. The comparison is made according to the architecture (SFC control and data planes) and the approaches' performance. The key points of comparison in the SFC control plane are:

- Implementation: Shows the technologies used to implement the SFC solution's control plane.
- SFP Adjustment: A dynamic SFP computing in the run-time phase with an approach such as SFP-fail over, SFP with better latency, traffic engineered SFP, and SF/SFP load balancing.
- Orchestrator-based: Shows if the SFC approach's control plane depends on an orchestrator or not.
- QoS/Policy Engine: Shows if the SFC solution's control plane has the capability of enforcing QoS and policies into the network.

The key points of comparison in the SFC data plane are:

- SFF: Explores the scheme applied by the SFC solution on the SFFs in order to steer traffic through the chains.
- SFC Classifier: Shows how to classify incoming traffic.

The key points of comparison in the approaches' performance are:

- Flexibility: Shows the level of flexibility in the SFC approach. The flexibility level is based on the efficiency of the traffic steering scheme implemented in the SFC solution.
- Scalability: Defines the level of scalability in the SFC approach. The scalability level is based on the number of rules needed to apply traffic steering for one chain.

SDN-based SFC Solutions

In [3], the NIMBLE system proves the potential of SDN to simplify and improve the existing middle-box management deployments, addressing challenges relevant to middle-box composition, load balancing, and packet modifications. The proposed NIMBLE system permits network operators to abstract the logical view of the middle-box policy and automatically pushes the forwarding rules into the switches. It considers the network topology, switches' capacities, and middle-box resource constraints. The NIMBLE design implies three main ideas. The first idea consists of the support of the middle-box composition by an efficient data plane that has tunnels between switches and pushes tags to packet headers using the SDN capabilities in order to know the processing status of each packet. The second idea is to provide resource management in a practical unified way and optimization using information on the switches' capacities and load balancing based on traffic fluctuations. The third key idea is to let the middle-box act dynamically by reporting the capabilities of SDN switches to design lightweight flow correlation schemes. A proof-of-concept of NIMBLE is showcasing the improvements achieved in terms of middle-box load balancing. The results also demonstrated the speed of the network bootstrap, and the high responsiveness of the system to network dynamics and load rebalancing.

In [4], a Squid-based FlowTags architecture is proposed whereby middle-boxes add tags to transmitted packets to communicate the necessary middle-box context (e.g., source hosts or internal cache state). Switches and middle-boxes can utilize these tags to provide consistent policy enforcement. An SDN controller is responsible for pushing the actions to the switches and middle-boxes in order to use the proposed tags in the packet header. FlowTags modify the architecture of the interface between the controller and switches by providing a new southbound interface for the flow tagging configuration process and for communication establishment with the FlowTags-aware middle-boxes. The modification takes part in three dimensions. First, FlowTags-aware middle-boxes are assumed to have the ability to process the incoming tags and add new tags based on the context. These tags are used by switches to steer the traffic. Second, a new FlowTags interface is proposed between the SDN controller and the FlowTags-aware middle-boxes. Third, a new control application is assumed to be used for the configuration of tags at switches and middle-boxes that is ultimately for the enforcement and verification of policies. In [4], the authors also provided a proof-of-concept

implementation to show how they modified Squid to support FlowTags and to also demonstrate the capability of a new policy enforcement process. This work seems promising, but there are still significant challenges to tackle. These challenges are related to the scalability and flexibility of the approach.

The position paper in [5] introduced a high-level concept and architecture to support SFC based on OpenFlow in a telecommunication network environment. The architecture supports the assignment of multiple subscribers to a single service while conserving the desired information about subscriber identification by the SF. The proposed model facilitates the SF instances deployment operation by reducing the network configuration modifications needed during deployment. In addition, the architecture instantiates many SF instances with low overhead. The SF instances are dedicated to one SFC only at a time, which provides an isolated network environment. Therefore, the model does not need to do packet matching conditions at the classifier. The separation of SF instances avoids using a consistent network addressing approach that crosses the whole service chaining system. The forwarding of traffic between the SF instances and switches is MAC address based. A proof-of-concept implementation for a relevant use case was provided to evaluate the feasibility of the model.

The work in [6] provides a proof-of-concept implementation of the SDN-based SFC approach presented in [5]. The approach merges common SFs without knowing their chain details. The proposed conception of SF instance separation facilitates the instantiation of SFs and provides a high degree of flexibility. The prototype's feasibility is tested over a hardware device that hosts a group of SF instances. These SFs were used to instantiate SFC for two types of applications (web traffic and video streaming). The demonstration showed the dynamicity of allocating users to new service classes.

A service-oriented SDN controller is proposed in [7] that deploys a programmable data delivery route by setting up multiple chains of VNFs existing in different locations of an OpenFlow-enabled network within the framework of service overlay networks. It also provides network service control, orchestration, and SDN network control functions in order to cope with the "extended QoS" requirements, and provides context-aware delivery of application service data. Moreover, the SFC context-aware architecture provides a realistic differentiation feature known as class-based forwarding. This feature simplifies the scalability issues, resulting in a decreased number of flow entries installed in the network switches. The authors have validated their proposed controller experimentally. The results proved that network optimization could be reached by assigning a specified number of crossed SF instances.

The model proposed in [8] presents a software architecture to dynamically instantiate network function-flow graphs (NF-FGs) beginning from a high level description of the targeted graphs and the existence of a specific incident (e.g., a new user is connected to the node), ending with common traffic steering provided by the SDN architecture. SDN technology is used to dynamically reset the network paths inside the network unit. Traffic forwarding among the nodes of the NF-FG is based on eXtensibleOpenFlowDatapath daemon (xDPd). xDPd is a software switch that creates multiple software Openflow switches in a dynamic way, called logical switch instances (LSIs). These LSIs can be tied together to physical interfaces and to NFs. A three-fold process occurs when the orchestrator node receives a new NF-FG description. First, it calls each required NF implementation and installs it. Second, it instantiates a user-LSI on xDPd, and then attaches it to the suitable NFs and to the classifier. Third, it instantiates an OF controller per each tenant that provides the insertion of appropriate rules into the flow tables of the LSIs.

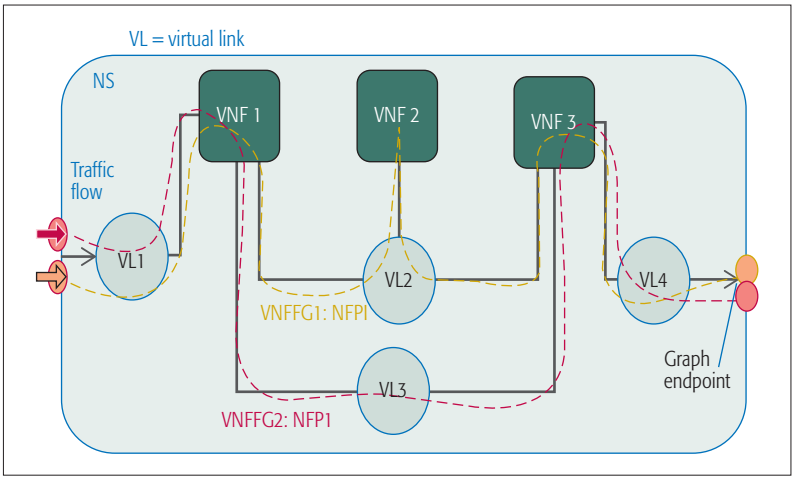

Figure 3. NS with two VNFFGs with different NFPs (ETSI GS NFV-MAN 001 V1.1.1).

The StEERING framework was introduced in [9] to provide an SFC model supporting dynamic traffic routing. StEERING uses a simple central controller that can adjust the traffic steering of various flow types through the targeted chain of middle-boxes. Moreover, it supports high scalability at the level of users and application policies. Scalability is offered through three dimensions. First, the rules at switches can be scaled linearly with the number of users and applications by using multiple tables to convert a single policy space into a multi-dimensional space. Second, it facilitates the integration of various types of policies by specifying the ordered group of service functions that each flow crosses as one type of metadata, so every table can work on the service functions separately. Third, the model provides the classification and header editing rules at the gateways only once within the network. The authors have provided a prototype to check the feasibility of their implementation and show its efficiency in providing flexible routing.

SIMPLE [10] is an efficient routing model for connecting SF instances and an approach to load balancing the SF instances. SIMPLE explicitly considers the inclusion of legacy SF instances. SIMPLE permits allocation of a logical middle-box steering policy and directly transposes this into forwarding rules that consider the network topology, switch capacities, and SF instances resource constraints. In the SIMPLE design, a particular SF instance is chosen to run within the limits of existing SDN capabilities (e.g., OpenFlow) and there is no need

Management of resource utilization is also required in the SFC framework to ensure high-speed communication for delivering ready-to-use media-optimized applications in SDN networks [17]. Such features are deemed important to enhance QoS provisioning to the users and applications as well.

to reconfigure SFs implementations. This article provides an approach to track packets when processed by SFs that modifies the packet header information. The approach relies on correlating packets before processing by an SF instance and afterward, which does not require modifications or even detailed knowledge of the SF instance. However, the approach needs the system to collect packets for the correlation analysis. In addition, the approach is rather complex as it needs packet matching with high accuracy to perform the classification function.

SDN and NFV-Based SFC Approaches

The MIDAS architecture is proposed in [11] to solve the problems of simultaneously detecting middle-boxes and selecting among multiple network function (NF) providers. MIDAS is based on a central controller per each NF provider to support coordination of traffic steering installation among all NF providers. MIDAS has the capabilities of middle-box signaling, controller chaining, and multi-party computation (MPC), which support on-path installing setup. MPC is used for NF provider assignment. MPC is characterized by privacy conservation, so it is used for middle-box usages over the NF providers. The MIDAS architecture featured with multiple NF providers cooperates for consolidated middle-box (CoMBs) detection over the traffic path and CoMB selection while preserving confidential information. The proposed architecture is based on three units: the CoMBs; a logical centralized controller per each NF provider; and the network processing client (NPCL) that provides the client's network service requests (CNSR). The authors also proposed a heuristic selection algorithm called the Intra-Provider Middle-box selection algorithm for NF allocations to CoMBs in the right place with an objective of load balancing provisioning over the CoMBs. They analyzed the applicability of MIDAS using the implemented prototype by delay measurement afforded during flow installing setup among all existing NF providers, middle-boxes, and CNSR arrival rates. The results showed that MPC does not have scalability problems as the MPC delay is not elongated by the CNSR arrival rate and the number of NF providers, which does not override the average internal path length. Simulation outcomes showed that utilizing MPC with the proposed middle-box selection algorithm shows good load balancing results and high request acceptance rates.

The ESCAPE prototype system, introduced in [12], is a developing and testing system for different nodes of the service function chaining framework. This model is applied to the UNIFY architecture. It is based on Mininet, Click, POX, and NetCONF tools integrated together in the ESCAPE framework. In addition, an orchestrator layer is added to allow SFC configuration, allocating VNFs into the physical resources, flow routing across the VNFs based on policies, and provisioning live management information on operating VNF instances. ESCAPE adopts VNF deployment by implementing a simple Mininet-based API where chain paths are constructed from available VNFs. These VNFs can be deployed and examined automatically. Moreover, a compact set of VNFs implemented in Click constructs a VNF catalog inside the ESCAPE system. The article presents a demo to show each unit of the architecture in a joint GUI. The demo includes:

- VNF containers and topology specification.
- Usage of a service graph to create chains.
- Service graph allocation to network resources.
- Traffic generation using standard tools.
- Monitoring the VNFs using Click.

The work in [13] introduced a new architecture that provides policy-based network management and has the ability to orchestrate and simplify fast deployment of various VNFs within an SDN/NFV environment. The architecture allows the selection of VNFs from available VNF instances using a policy engine staying in the NFV orchestrator. This NFVO provides various stitched VNFs, using them to build OSS/BSS applications. Moreover, the architecture addresses VNF life cycle management and service chaining among these different VNFs sent to large scale customers. The proposed architecture features:

- The ability to separate hardware elements, VNFs, services, and orchestration.
- Abstraction of network resources and network functions through predefined information models.
- Policy-based management allowance for singular VNFs and orchestration of NFV service chains.
- The ability to deploy NFV services ruled by policies.

The authors have deployed a prototype to provide an evaluation for their proposed architecture. They presented the use case of a telecom operator who instantiates VNFs on-desire for the management of network traffic outgoing from the content delivery network (CDN) caching nodes of CDN providers positioned inside the operator's sites. They implemented a policy-based traffic engineering service by supporting VNF deployment, virtual links assignment to the physical topology, flow monitoring, and orchestration.

The authors in [14] focus on SFC implementation in a cloud-based edge data center network where all SFs are software applications operating in virtual machines within these data centers. The main target of this work is to prove that this new softwarized environment permits a high level of flexibility and dynamicity of SFC in comparison with the traditional hardware-based architectures. To reach these flexible and dynamic SFCs for Layer 2 and Layer 3 edge network function implementations, the SDN control plane is used to provision the forwarding rules into OpenFlow switches. They also provided a proof-of-concept using Mininet emulation in order to evaluate their approach under a feasible scenario. The results showed that they can provide dynamic SFC and flexible traffic routing.

The work in [15] shows how telecom operators benefit from the NFV and SDN paradigms to improve the management of SFs and construct new business models. The article targets two major sides. The first side is how telco infrastructure deploys this new paradigm. The second issue is orchestration and management of SFs in distributed telco cloud environments by presenting the Cloud4NFV platform. The approach of modeling SFs in the cloud infrastructure is highlighted in that work, and the ability to perform SFC provisioning

SFC solutions	Architecture						Performance	
	SFC control plane				SFC data plane			
	Implementation	SFP adjustment	Orchestrator based	QoS/policy engine	SFF	SFC classifier	Flexibility	Scalability
NIMBLE [3]	SDN	Dynamic SFP with load balancing approach			Tags based	Packet matching	Low	Medium
FlowTags [4]	SDN	Static SFP		✓	Tags based	Packet matching	Low	Medium
SDN-based SFC [5, 6]	SDN	Static SFP			MAC address based	N/A	Medium	Medium
Context-aware SFC [7]	SDN	Static SFP		✓	MAC address based	Class-based forwarding	Medium	High
User-specific SFC [8]	SDN	Dynamic SFP	✓		xDPd	Packet matching	Low	Low
StEERING [9]	SDN	Dynamic SFP		✓	MAC address based	Packet matching	Medium	High
SIMPLE [10]	SDN	Dynamic SFP with load balancing approach		✓	MAC address based	Complex packet matching	Medium	Low
MIDAS [11]	SDN & NFV	Static SFP with load balancing approach			MAC address based	N/A	Medium	Low
ESCAPE [12]	SDN & NFV	Dynamic SFP	✓		MAC address based	Policy based	High	Medium
Policy-based SFC [13]	SDN & NFV	Static SFP	✓	✓	N/A	Policy based	High	High
Cloud-based SFC [14]	SDN & NFV	Dynamic SFP			MAC address based	Packet matching	Medium	Low
Cloud4NFV [15]	SDN & NFV	Static SFP	✓		N/A	Packet matching	High	High
Optical SFC [16]	SDN & NFV	Static SFP	✓		Optical circuit switches	N/A	High	High

Table 1. Taxonomy of the prior research work relevant to the SFC concept and its implementations.

is demonstrated as one of the essential features of SF composition. The Cloud4NFV platform is constructed over cloud, SDN, and WAN technologies to provide SF as a service. The Cloud4NFV platform also provides service monitoring and deployment, and optimized WAN and cloud resources for SFs support. A proof-of-concept is presented to evaluate some practical examples of the possible advantages of the proposed platform and the given standards in a telco environment.

In [16], an optical SFC architecture is proposed. The proposed architecture steers functionality into the datacenters for SFC using wavelength switching. The authors set up a packet/optical hybrid datacenter architecture to steer large volumes of flows in an optical steering network. They introduced such a solution to cope with the limitations of packet-switched SFC, such as complicated configuration of flow matching rules when the number of flows increases, which may lead to high operational cost, inefficient power consumption, and performance degradation due to scalability. The architecture consists of an operations support system/business support system (OSS/BSS) module, connected to an SDN controller and a NFV manager. The SFC configuration is done at the OSS/BSS module. Furthermore, the OSS/BSS module enforces the operator's policies. The SDN controller and NFV manager are responsible for resource allocation. The optical steering layer, including the network nodes, is placed on the southbound side of the SDN controller, which uses the OpenFlow v.1.4 protocol with an extension for optical circuit configuration to communicate with the optical circuit switches in the data plane layer. The proposed architecture shows its advantages, compared to packet-based routing, in terms of flexibility, scalability, reduced operational complexity, and energy efficiency.

Comparison and Evaluation

Table 1 shows some of the taxonomy used in the above mentioned approaches. The taxonomy shows that the approaches that adopt SDN and NFV technologies together alongside the orchestrator layer provide higher SFC scalability and flexibility than others. This comparison shows that most of the SFC approaches did not involve QoS and policy enforcement and neglect the load balancing functionality. Most of the frameworks use MAC address and OpenFlow functionality to apply traffic steering among the SFs without NSH support, as specified by the IETF SFC group. The usage of MAC address and/or OpenFlow protocols without NSH support has limited scalability and is more complex than using them with

There are two standards for SFC: one by IETF SFC WG and one by ONF. These standards impose the requirements that should exist in each SFC architecture, design, or implementation. These requirements are used to define the gaps and limitations in the previous SFC-related research work.

NSH support. There are some approaches that use tags instead of NSH. The work presented in [16] defines a solution to this limitation that applies optical steering of the data plane, using optical circuit switching devices that enhance the scalability and flexibility in the SFC domain network.

Challenges and Limitations

This section highlights the common limitations in the previous work and summarizes the open challenges relevant to the SFC concept and architecture. NSH capability in switches is one important challenge. Indeed, there is a lack of NSH-supported switches. As a countermeasure, some previous research works consider instead the use of tags or MAC addresses to steer traffic among the SFs. The trend is also toward supporting NSH in virtualized switches such as Open vSwitch (OvS). SFs do not have NSH capability either. Consequently, an SFC proxy must be used to encapsulate and de-encapsulate the packets travelling to and from SFs. However, the SFC proxy process may impact network performance, which can be alleviated only by equipping SFs with NSH support.

Traffic-engineered (TE) SFC is needed to provide an optimized SFC network with short computational latency. The literature provides limited concepts of traffic engineering in SFC. Some research works provide QoS-aware SFC paths to meet user and application requirements; other research works aim at maximizing the available data rate on the network links or cost savings. TE SFC needs to support all these features. This will be possible only by improving the SFC architecture through a well synchronized monitoring system to collect the required information from the network, QoS probes to test the reliability and performance of the existing SFPs, and a TE system that has the ability to instantiate TE-SFPs when needed. In terms of programmability, an efficient scheme is required to provide the optimized TE-SFP that satisfies the QoS requirements and network performance requirements.

The placement of SFs is a challenge and not sufficiently investigated in the literature. Furthermore, to the best knowledge of the authors, it was never investigated in the case of a network bottleneck scenario. There are two options in this scenario. The first option is to migrate the SF instance to a new location in the network; the second option is to instantiate a new SF instance. The choice between the two options adds a new challenge and must be investigated. The best location for the migrated or new instantiated SF must also be investigated, and novel optimal placement schemes must be proposed. There is also no previous work in the literature that discusses the use of SFC under different SLAs to support different classes of service.

Management of resource utilization is also required in the SFC framework to ensure high-speed communication to deliver ready-to-use media-optimized applications in SDN networks [17]. Such features are deemed important to enhance QoS provisioning to users and applications as well.

Conclusion

The delivery of end-to-end service requires various service functions to be provisioned in a SFC. This article introduces a survey of all existing SFC architectures, and conceptual approaches that are based on SDN and NFV. Research works are presented, compared, and evaluated. There are two standards for SFC: one by IETF SFC WG and one by ONF. These standards impose the requirements that should exist in each SFC architecture, design, or implementation. These requirements are used to define the gaps and limitations in the previous SFC-related research work.

Finally, the open challenges are discussed. Policy-based SFC, Cloud4NFV, and Optical SFC architectures exhibit high performance in terms of SFC orchestration, scalability, and flexibility to provide SFC in cloud environments, making use of SDN and NFV technologies.

Acknowledgment

This work has been funded with the support of the European Commission. This article reflects the view of the authors only. The European Commission cannot be held responsible for any use that may be made of the information contained therein.

References

[1] W. John *et al.*, "Research Directions in Network Service Chaining," *IEEE SDN for Future Networks and Services (SDN-4FNS)*, 2013, pp. 1–7.
[2] T. Taleb *et al.*, "Coping with Emerging Mobile Social Media Applications Through Dynamic Service Function Chaining," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, 2016, pp. 2859–71.
[3] Z. Qazi *et al.*, "Practical and Incremental Convergence between SDN and Middleboxes," *Open Network Summit*, Santa Clara, CA, 2013.
[4] S. Fayazbakhsh *et al.*, "FlowTags: Enforcing Network-wide Policies in the Presence of Dynamic Middlebox Actions," *Proc. 2nd ACM SIGCOMM Wksp. Hot Topics in Software Defined Networking*, 2013, pp. 19–24.
[5] J. Blendin *et al.*, "Position Paper: Software-Defined Network Service Chaining," *3rd European Wksp. Software Defined Networks*, 2014, pp. 109–14.
[6] J. Blendin *et al.*, "Demo: Software-Defined Network Service Chaining," *3rd European Wksp. Software Defined Networks*, 2014, pp. 139–40.
[7] B. Martini *et al.*, "SDN Controller for Context-Aware Data Delivery in Dynamic Service Chaining," *1st IEEE Conf. Network Softwarization (NetSoft)*, 2015, pp. 1–5.
[8] I. Cerrato *et al.*, "User-Specific Network Service Functions in an SDN-enabled Network Node," *3rd European Wksp. Software Defined Networks*, 2014, pp. 135–36.
[9] Y. Zhang *et al.*, "Steering: A Software-defined Networking for Inline Service Chaining," *21st IEEE Int'l. Conf. Network Protocols (ICNP)*, 2013, pp. 1–10.
[10] Z. A. Qazi *et al.*, "SIMPLE-fying Middlebox Policy Enforcement Using SDN," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 43, no. 4, 2013, pp. 27–38.
[11] A. Abujoda and P. Papadimitriou, "MIDAS: Middlebox Discovery and Selection for On-path Flow Processing," *7th Int'l. Conf. Communication Systems and Networks (COMSNETS)*, 2015, pp. 1–8.
[12] A. Csoma *et al.*, "ESCAPE: Extensible Service Chain Prototyping Environment using Mininet, Click, Netconf and POX," *ACM SIGCOMM Computer Commu. Rev.*, vol. 44, no. 4, 2015, pp. 125–26.
[13] K. Giotis, Y. Kryftis, and V. Maglaris, "Policy-based Orchestration of NFV Services in Software-defined Networks," *1st IEEE Conf. Network Softwarization (NetSoft)*, 2015, pp. 1–5.
[14] F. Callegati *et al.*, "Dynamic Chaining of Virtual Network Functions in Cloud-based Edge Networks," *1st IEEE Conf. Network Softwarization (NetSoft)*, 2015, pp. 1–5.
[15] J. Soares *et al.*, "Toward a Telco Cloud Environment for Service Functions," *IEEE Commun. Mag.*, vol. 53, no. 2, 2015, pp. 98–106.
[16] M. Xia *et al.*, "Optical Service Chaining for Network Function Virtualization," *IEEE Commun. Mag*, vol. 53, no. 4, 2015, pp. 152–58.
[17] F. Pop *et al.*, "Adaptive Scheduling Algorithm for Media-optimized Traffic Management in Software Defined Networks," *Computing*, vol. 98, no. 1–2, 2016, pp. 147–68.

Biographies

Ahmed M. Medhat (a.hassan@campus.tu-berlin.de) is currently a Ph.D. candidate and a research assistant on the faculty of electrical engineering and computer sciences, Technical University of Berlin, Germany. He received his B. Sc. degree in information engineering and technology, and his M.Sc. degree in communication engineering from German University in Cairo (GUC), Egypt in 2010 and 2011, respectively. His research interests are in the field of next generation network infrastructures (related topics to SDN and NFV) with a focus on service function chaining and its quality of service enhancement solutions.

Tarik Taleb [M] (talebtarik@ieee.org, tarik.taleb@aalto.fi) is currently a professor at the School of Electrical Engineering, Aalto University, Finland. He has worked as a senior researcher and 3GPP standards expert at NEC Europe Ltd. Prior to his work at NEC, until March 2009, he worked as an assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI. He received his B.E. degree in information engineering with distinction, and his M.Sc. and Ph.D. degrees in information sciences from Tohoku University in 2001, 2003, and 2005, respectively. His research interests lie in the field of architectural enhancements to mobile core networks (particularly 3GPP's), mobile cloud networking, mobile multimedia streaming, and social media networking. He has also been directly engaged in the development and standardization of the Evolved Packet System. He is a member of the IEEE Communications Society Standardization Program Development Board and serves as Steering Committee Chair of the IEEE Conference on Standards for Communications and Networking. He has received many awards for his many contributions in the area of mobile networking.

Asma Elmangoush (asma.elmangoush@alumni.tu-berlin.de, asma.a.elmangoush@campus.tu-berlin.de) received her B.E and MS.c. in computer engineering from the College of Industrial Technologies-Misurata, Libya, and received her Ph.D. degree from the Technical University Berlin in 2016. She is currently a lecturer at the College of Industrial Technologies-Misurata, Libya.

Giuseppe A. Carella (giuseppe.a.carella@tu-berlin.de) is a senior researcher at the Fraunhofer FOKUS and at the Technische Universität Berlin (TUB). He received his M.Sc. in engineering of computer science from the Alma Mater Studiorum University of Bologna in 2011. During his studies he focused on next generation network infrastructure, especially in IMS services, such as presence and messaging. In 2012 he joined the Next Generation Networks (AV) team at the Technical University Berlin, where he started investigating topics related to SDN and NFV in the context of his Ph.D. studies. His strong background in cloud computing is the basis of his research and contributed to the virtualization of the software-based network functions developed at Fraunhofer FOKUS, namely OpenEPC and Open5GCore. He is currently leading the team developing the Open Baton toolkit, an open source platform providing the means for building a comprehensive NFV environment.

Stefan Covaci (stefan.covaci@tu-berlin.de) is a senior solutions architect for future Internet services platforms on the computer sciences and electrical engineering faculty of Technical University of Berlin, Institute for Telecommunication Systems. Between 1990 and 2007 he was working at GMD FOKUS (today Fraunhofer Institute FOKUS) acting as director of the competence center Intelligent Mobile Agents. He co-initiated and participated in the agent standardization work of OMG (Object Management Group) and FIPA (Foundation of Intelligent Physical Agents), and was one of the co-authors of the first agent technology standard, the OMG-MASIF (Mobile Agent System Interoperability Facility). He has extensive experience in the management of a variety of projects and study contracts for the European Commission, German Agencies (BMBF, BMWF), and other industrial national and international organizations in the areas of IT and telecommunication networks and services. His recent work is in the areas of next generation network infrastructures and service delivery platforms for the Future Internet, with a focus on interoperability and management solutions. He is member of the Architectural Board of the European Future Internet Public-Private Partnership (FI-PPP), and technical coordinator of the FI-STAR project, applying Future Internet technology in the healthcare sector. He is the technical coordinator of the DAAD–UNIFI project enabling future Internet academic teaching and research in developing countries. He has published more than 90 papers and was a chair or member of the program committee of many international conferences.

Thomas Magedanz [SM] (magedanz@ieee.org, Thomas.magedanz@fokus.fraunhofer.de) is a full professor on the electrical engineering and computer sciences faculty at the Technical University of Berlin, Germany, leading the chair for Next Generation Networks. In addition, he is the director of the "Software-based Networks" division of the Fraunhofer Institute FOKUS. In 2006, he was named an Extraordinary Professor in the Department of Electrical Engineering of the University of Cape Town, South Africa. Since 2007, he has also been a visiting professor in the Department of Mathematics, Physics and Computing at the Waterford Institute of Technology in Ireland. For more than 20 years he has been working in the convergence field of fixed and mobile telecommunications, the Internet, and information technologies, which resulted in many international R&D projects centered around next generation service delivery platforms prototyped in a set of globally recognized open technology testbeds. In 2007 he joined the European FIRE (Future Internet Research and Experimentation) Expert Group. In the course of his research activities he has published more than 250 technical papers/articles. In addition, he is a senior member of the IEEE, and serves on the editorial board of several journals. He received his diploma and his Ph.D. in computer sciences from the Technical University of Berlin, Germany, in 1988 and 1993, respectively. In 2000 he finished his postdoctoral lecture qualification in applied computer sciences at the Technical University of Berlin, Germany.