A Survey on Service Function Chaining

Deval Bhamare[Ψ], Raj Jain[¥], Mohammed Samaka[Ψ], Aiman Erbad[Ψ]
[Ψ]Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[¥]Department of Computer Science and Engineering,
Washington University in St. Louis, St. Louis, MO 63130
{devalb, samaka.m, aerbad}@qu.edu.qa,
jain@wustl.edu

***Abstract*— Cloud computing is gaining significant attention and virtualized datacenters are becoming popular as a cost-effective infrastructure. The network services are transitioning from a host-centric to a data-centric model moving the data and the computational resources closer to the end users. To meet the dynamic user demands, network operators have chosen to use elastic virtual resources to implement network services over static rigid physical model. With the advent of network function virtualization (NFV), network services instances are provisioned across multiple clouds for performance and load balancing purposes. Interconnection of these instances to form a complete end-to-end network service is complex, time consuming and expensive task. Service function chaining (SFC) is a mechanism that allows various service functions to be connected to each to form a service enabling carriers to benefit from virtualized software defined infrastructure. SFC is an enabler for NFV, providing a flexible and economical alternative to today's static environment for Cloud Service providers (CSPs), Application Service Providers (ASPs) and Internet Service Providers (ISPs). This paper provides a closer look at the current SFC architecture and a survey of the recent developments in SFC including its relevance with NFV to help determine the future research directions and the standardization efforts of SFC. Finally, the paper discusses open research topics in relevance with the SFC architecture and demonstrates a need for an analytical model for the SFC architecture to achieve the optimal performance.**

***Index Terms*— Service function chaining, Network function virtualization, Optimal placement, SDN, Survey, Multi-cloud.**

I. INTRODUCTION

In recent years, there is a trend to virtualize the network, the storage and the computational resources. However, the underlying network is still mostly physically managed. Network operators are currently struggling to meet growing user and network traffic demands on their traditional networks. While end users want constantly declining "cost per bit", Internet service providers' (ISPs) capital expenditures (CAPEX) and operational expenditures (OPEX) for increasingly complex network infrastructure are rising [1, 5]. Also, recently there has been an exponential growth in the user traffic due to the explosion of mobile devices and the emergence of novel networking paradigms such as the Internet of Things (IoT) [74]. Under such situations network operators may benefit from Network Function Virtualization (NFV) model, with which the network services can be deployed as virtualized services [4, 5].

Due to the advancements in the field of cloud computing, computational and storage resources are getting deployed across geographically distributed areas, bringing such resources closer to the end-user bases. In such situations, application service providers (ASPs) may improve the end-user experience by deploying their services across multiple datacenters spread across multiple, geographically distributed clouds [1, 2]. However, the current underlying network model is static, rigid, and lacking auto-configuration abilities. In such cases, network operators may have to face several challenges with the traditional ISP networks. These challenges include dependence on the physical topology, complex time and resource consuming operations (for example, adding, deleting or updating services), static path provisioning, and manual load balancing among others [14]. However, recent technologies such as software defined networking (SDN), network function virtualization (NFV) provide operators with the tools to tackle existing challenges. Recent advancements in the field of SDN and NFV allow ISPs to deploy many network services over the virtualized infrastructures [4, 24, 58, 65, 89]. SDN especially allows flexible and efficient network forwarding among the virtual network functions (VNFs). These VNFs include middle-boxes like firewalls, load-balancers, proxy servers, deep packet inspectors (DPIs), intrusion detectors and others, including core telecom stack [119].

Advancements in the field of SDN have lead ISPs to move to a novel architecture using network function virtualization (NFV) [13, 27, 65, 75]. SDN may prove effective in improving some aspects of the ISP networks such as security, quality of service (QoS) and service level agreements (SLAs) [89-91, 109]. Though NFV along with SDN provides flexibility to ISPs for deployment of their network services; the creation, deletion and interconnection of these virtual functions is a challenging task due to the dynamic end-user demands and the network parameters, and the complexity of the virtual function addition/deletion. In addition, virtual functions may need to be executed in a specific order to provide a complete end-to-end network service. The order of execution depends on the type of user requests and is highly dynamic in nature [7, 9]. For an effective deployment of these virtual functions and scalability of the NFV architecture, there is a need for mechanisms, which can automatically form the ordered chain of such functions, dynamically guiding the user requests through such chains. Without such mechanisms, interconnection of these virtual network functions is ad-hoc and error-prone task adding to the administrative complexities and costs of the existing networks and resulting in high OPEX for the ASPs and CSPs [15, 69].

E-preprint
Journal of Network and Computer Applications, Vol. 75, Nov. 2016, pp. 138-155

Introducing a new network service or updating the existing one is a complicated, time-consuming and expensive task for the network operators [68, 69]. This rigidity complicates reconfiguration of the contemporary network [4]. Recent advancements in network virtualization have made it possible for many network services to be implemented as virtual functions such as: NATs, Firewalls, Deep Packet Inspectors (DPI) and many more [91-93]. The physical infrastructure may be shared by these network services. The technology is known as Network Function Virtualization (NFV) [40-43]. NFV is a great tool to the network operators for more organized networks [64, 66]. However, proper interconnection of service functions is necessary for the proper flow of the packets in the network. Informally, interconnection of two or more service functions in the network, for a complete end-to-end service, is known as "*Service Function Chaining*" or simply SFC.

With the recent explosion in mobile devices, and sensory devices due to technologies such as Internet of Things (IoT) [73, 83], service providers may benefit from deploying their applications across multiple clouds, so that the storage and the computational power can be brought closer to mobile user-bases [67, 68]. This may reduce the total response time to the end users and cost to the ASPs by reducing the use of expensive WAN links. Service function chaining (SFC) enables proper interconnection among the virtual functions (or the service functions) spread across multiple clouds. Currently, SFC is at a nascent stage of development. There are many challenges, which need to be addressed by ISPs and CSPs in terms of service function chain deployment to maximize benefits of the NFV technology [64, 65, 95].

Currently, CSPs and ISPs need to manually form ordered chain of physical middle-boxes, such as firewalls, NATs, etc. They are expected to make sure that the service chains meet the policy constraints, the computational and the network capacity constraints as well as have acceptable total latency for end-users. On the fly changes need to be done to these service chains, such as deletion or addition of new virtual functions depending on user demands. In addition, allocation of resources and placement of network virtual functions need to be performed dynamically. It is a daunting task for the ISPs to scale the model in a dynamic network environment. Hence, there is a need for a well-tested, dynamic and automatic service chaining model to save the efforts, time and cost to the operators [5, 9]. In this article, we summarize the approaches that have been studied in the literature towards standardization and implementation of SFC. Broadly we classify the research for SFC into two domains as: (1) architectural models and implementation of SFC, and (2) optimization models for the network services and/or virtual functions distribution and allocation. Various models have been proposed in the literature which optimize different parameters such as cost, end-to-end latency, network traffic, network bandwidth, energy, overall resources required and others. Such models are necessary for the operators 1) to achieve the optimal performance of the networks, 2) to satisfy the user demands in a timely manner, 3) to accommodate the dynamic SLAs, and 4) to minimize the costs for the operators. We observe that there is a dearth of analytical studies and optimization models for such the SFC implementation approach. On the contrary, the optimal models suggested for placement of the virtualized network functions (VNFs) lack practical values and need modifications to suit to the SFC framework. There is a need for the combined study of these problems with the practical implementation and the analysis of the results.

The remaining of the paper is organized as follows. In Section II, we explain simple use cases of service function chaining and demonstrate the need for optimization. Section III describes the scope of SFC and its relevance with NFV. We also outline the major terminology defined in the standards to better understand the SFC discussion in subsequent sections. Section IV describes the SFC architecture. We discuss the major developments in the field of SFC and its practical implementations as a major enabler for NFV. We also have a closer look at the various architectures proposed by ETSI, IRTF, IETF and ITU-T. Section V describes the optimization strategies studied in the literature, which can be adopted to suit the SFC architecture. We argue this is an important step for efficient and flexible service chains. In Section VI, we discuss open research topics in the field of SFC and possible future research directions. Finally, we conclude the paper. We provide description for all the acronyms used in this article in Table 1.

TABLE I. ABBREVIATIONS USED IN THE ARTICLE

Acronym	Description
ASP	Application service provider
CAPEX	Capital expenditures
CDN	Content distribution network
CSDL	Cloud service declarative definition language
CSP	Cloud service providers
DC	Datacenter
DPI	Deep packet inspector
ETSI	European Telecommunications Standards Institute
IaaS	Infrastructure as a service
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
ISP	Internet service provider
IRTF	Internet Research Task Force
ITU-T	International Telecommunication Union Telecommunication Standardization Sector
MPI	Message passing interface
NAT	Network address translator
NF	Network function

NFF	Network function forwarder
NFV	Network function virtualization
NFVI	Network function virtualization infrastructure
NFV-MANO	NFV Management And Orchestration
NMS	Network management system
NP	Nondeterministic polynomial
NS	Network service
NSH	Network service header
NVF	Network virtual function
OA&M	Operations, administration and management
OF	OpenFlow
OPEX	Operational expenses
PaaS	Platform as a Service
PM	Physical machine
QoS	Quality of service
SDL	Service description language
SDN	Software defined networking
SF	Service function
SFC	Service function chaining
SFCC	Service function chaining controller
SFCR	Service function chaining router
SFF	Service function forwarder
SFP	Service function path
SLA	Service level agreement
SMI	Service measurement index
SWA	Software architecture
USDL	Unified service description language
VF	Virtual function
VIM	Virtualized Infrastructure Manager
VM	Virtual machine
VNF	Virtual network function
VNFC	Virtual network function component
VNFFG	VNF Forwarding Graph
VNFM	VNF Manager
WSDL	Web service description language

II. SERVICE FUNCTION CHAINING: USE-CASE

In this section, we present simple use-cases of the service function chaining model. Then, we discuss the relevance of SFC with NFV and demonstrate the need for the optimal service function placement for the best use of the network capacity and for minimum response time. Finally, we demonstrate the need for various optimization models and an exhaustive analytical study of the SFC architecture. Optimization models may optimize various network parameters, such as, total allocated bandwidth, total end-to-end delays or total deployment cost. It is important to note that the examples presented are just illustrative examples, with many other optimization problems still open for SFC. Also, the scope of SFC is not only limited to the network services. SFC architecture is equally important for the transport services, multimedia services as well as application services.

Maintaining these service chains manually, however, is a tedious, expensive and error-prone task due to its dynamic nature. Other challenges include optimal placement of service functions to reduce the delays or/and minimize the required network capacity, allocation of the user demands to the service functions, and routing of traffic through different service functions with a focus on resource allocation. Scattered and multiple instances of service functions mandate user requests to travel through various service functions forming dynamic service chains [8, 9]. The SFC model needs to be flexible enough to accommodate the dynamic user demands and service policies. For example, users may be mobile and their position can change rapidly (such as a user travelling in a car while using the service), and the policy can vary based on latency so stringent latency constraints is applied on a video service. In addition, allocation of resources and placement of network virtual functions needs to be performed dynamically and automatically. Automation in SFC can save significant time and may result in significant OPEX saving for operators, such as reduction in administrative cost. However, such automation needs to be governed by optimization strategies to choose the optimal parameter values such as cost, end-to-end latency, network traffic, network bandwidth, energy, overall resources required and others for an improved end-to-end performance. It is important to note that optimization models may not scale for the larger networks and researchers will have to come up with the approximation algorithms to scale better. This will result in a carrier-grade SFC model which is reliable enabling high-availability and fault tolerance.

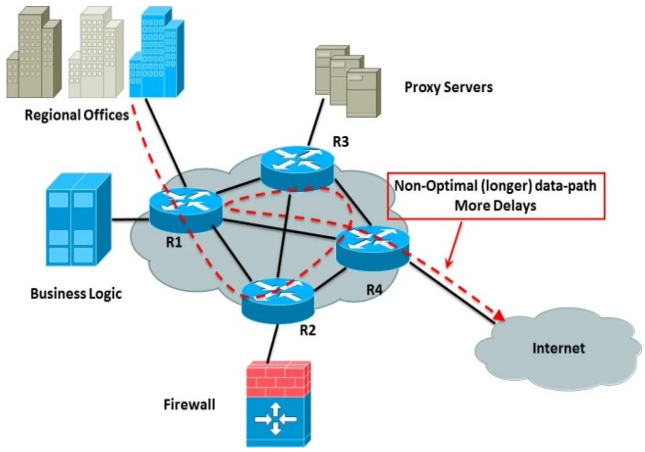

Figure 1. (a) Motivation: Non-optimal placement of Service Functions

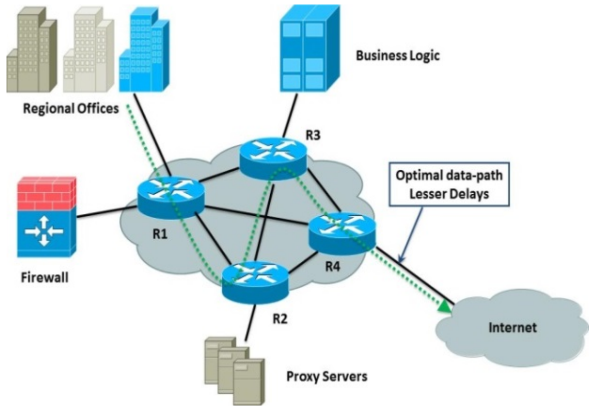

Figure. 1. (b). Motivation: Savings in network link capacities with optimal placement of Service Functions.

Figure 1 demonstrates with a simple example how link capacities and total delays to the end customers can be reduced with the optimal placement of service functions in the given network. Figure 1 shows a regional office for a hypothetical ASP, with its service available to users via Internet. The service consists of three virtual functions, that is, a firewall, a proxy-server and a business logic component in a given order. Depending on the deployment sites of these virtual functions, the traffic flows have to travel through different links and eventually different paths. This is a common case in the current network due to the distributed nature of the physical resources and the end users. First path (dashed red line in Figure 1-a) is a result of the non-optimal placement (maybe due to manual placement); however, the second path (dotted green line, Figure 1-b) is the optimal solution, saving link capacities and delays. Errors in service placement may induce more delays, complex reconfigurations and increased OPEX and CAPEX.

Let us now consider a case of a cloud service provider (CSP), providing a distributed micro-datacenter abilities for its mobile users. These micro-clouds, deployed across the edges, may have only web servers installed on them for quick updates to the users. This has led to the concept of micro-clouds at the cellular base stations [94]. This is becoming a common scenario in 4G/5G networks. ASPs and ISPs may benefit from lesser costs and better end-user experience. For example, an ASP such as Netflix, may store its favorite user videos at the micro-clouds implemented at the cellular base-stations for quick access to the end users. In such case, Netflix may also benefit by saving expensive WAN bandwidth. However, the database servers and resource-intensive computing servers are deployed on the clouds located at core locations. In such cases, the user request have to travel through multiple clouds to get fulfilled. For example, user from user-base 1 has to travel along the path represented by the solid black line (as it needs access to the database or the computing servers) and user from user-base 2 has to travel along the path represented by the dotted red line (as it needs quick response from the web server only), as shown in Figure 2.

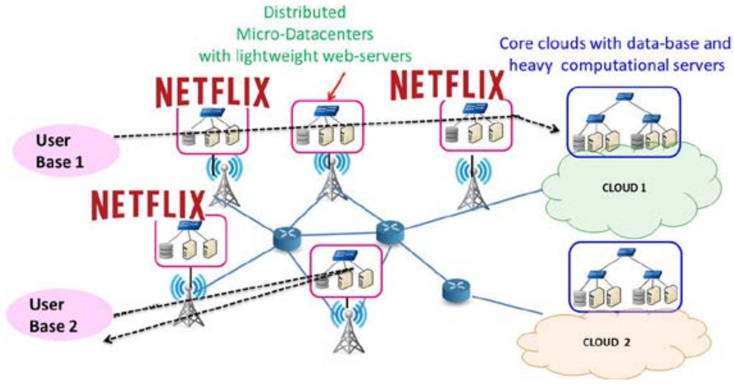

Figure. 2. SFC use case at Application level across multiple-clouds.

Path selection depends upon the type of services desired by the end-users. In such cases, proper service chains need to be formed and operated across multiple clouds at the application level. SFC may be a major enabler for the recent networking paradigm such as IoT, which is gaining popularity rapidly. IoT deals with a huge amounts of data (big-data) which needs to traverse through multiple applications or services [73, 79]. SFC has the potential to be an important tool for the dynamic steering of IoT-related big-data..

Being in the initial stage of the development, SFC faces several challenges, for example, dynamic service chain formation, optimal service function placement, enforcing policies and SLAs, physical topology independence, dynamic traffic steering and several others. Hence, there is a need for analytical model for SFC architecture, so that the nature of the complete system can be studied theoretically. The analytical models may help operators obtain close-to-optimal configurations. The contemporary static model is inherently bound to the underlying network topology [6, 8]. A small change in the user demand or a new service policy mandates the operators to change the topology, which they are hesitant to do due to the system complexity and the possibility of errors. Hence, automation with optimization is necessary to save the network operators from these hassles [48]. In the next section, we discuss the scope of SFC and its relevance with NFV for better insights.

III. NFV AND SFC: A PRIMER

Efforts to deploy the SFC architecture for dynamically created virtual services has gained significant traction recently in both research communities and standardization bodies [14, 40, 45, 72]. ISPs and ASPs seek to offer advanced services beyond the basic connectivity, while optimizing the infrastructure use and the operational efficiency. SFC architecture provides a service to the NFV model by interconnecting different virtual functions (or service functions) in a specific order [77, 79]. Hence, it becomes imperative to get familiar with the NFV model to better understand the challenges associated with the SFC. A brief look at the NFV architecture proposed by the European

Telecommunications Standards Institute (ETSI) is provided in this section.

ETSI has been working on standardizing the NFV model since 2012 [42]. As described by ETSI, "*Improved capital efficiencies and flexibility in deploying network services as compared to the dedicated hardware implementations are the major objectives of the NFV*". A commercial-off-the-shelf (COTS) hardware may be used to implement the service functions through software virtualization techniques [40]. This achieves scalability as well as independence from the underlying physical topology by allowing software and computing resources to be located at the most appropriate places [41]. Improved operational efficiencies, reduced power usage are among the many other objectives to be achieved with NFV. A significant amount of work has been done by ETSI for monitoring and orchestration of NFV. As a result, NFV MANO framework has been developed [105]. ETSI is also addressing the problem of connectivity among VNFs and proposing the use of VNF Forwarding Graph (VNFFG) to depict the chains of VNFs. Many organizations, vendors, operators, and service providers are currently working on this problem and have started proposing solutions. Open Source MANO (OSM), an ETSI-hosted project to develop an Open Source NFV Management and Orchestration (NFV-MANO) software stack aligned with ETSI NFV, is a good example of such efforts [105]. The VNF manager (VNFM) works in concert with other NFV-MANO functional blocks, such as virtualized infrastructure manager (VIM), to help standardize VNFs and increase the interoperability of these functions of virtual networking. A high-level NFV management framework given by ETSI is shown in Figure 3.

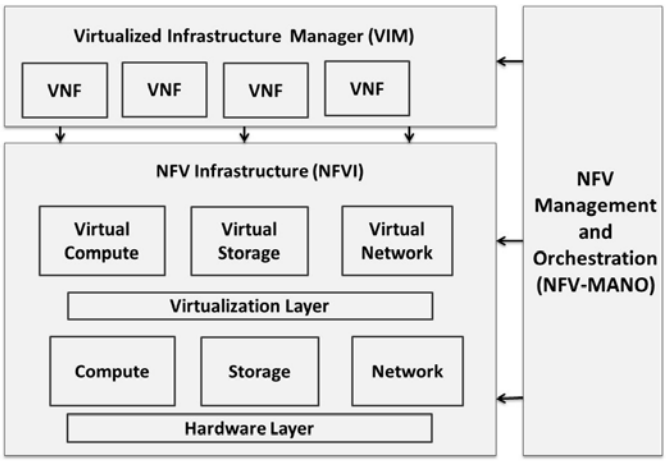

Figure. 3. High level VNFM architecture proposed by ETSI.

The major components of the NFV model include Virtual Network Functions (VNFs), or simply virtual functions (VFs), which are the software implementation of the network functions, such as a Firewall, a DPI and others. These functions run over Network Function Virtualization Infrastructure (NFVI). NFV management and orchestration module focuses on virtualization specific management tasks of the NFV architecture [40]. A framework for NFV comprising the policy architecture has been proposed by the IRTF in [72]. Other examples include projects lead by Metro Ethernet Forum (MEF) [106] and OpenStack [107]. Detailed discussion on NFV is out of scope of this article. For more detailed discussions on NFV, please refer to the works such as [45, 64-66]. With a quick review of NFV architecture we now focus on the advancements in the field of SFC. Internet Engineering Task Force (IETF) has taken the initiative towards standardization of the SFC architecture. In this paper, we adopt IETF's definitions for several important terms [14]. To better understand the proposed architecture by IETF in later sections, we first try to understand some of the important terms defined by IETF.

A. Service Function Chaining Definitions:

- *Network Service: Network Service is an offering provided by an operator that is delivered using one or more service functions. This may also be referred to as a composite service.* Network service is a complete, end-to-end functionality provided by the network operator such as "network protection system". A network service may comprise of one or more virtual functions or service functions, for example, a firewall, a deep packet inspection (DPI) and a virus scanner in the case of "network protection" system.

- *Service Function: A function that is responsible for specific treatment of received packets. A service function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). As a logical component, a service function can be realized as a virtual element or be embedded in a physical network element. One or more service functions can be embedded in the same network element. Multiple occurrences of the service function can exist in the same administrative domain.* If we consider the example of "network protection" system as a service provided by an ASP, as explained above, then its components, that are, the firewall, the DPI and the virus scanner would be its components and called as service functions.

- *Service Function Chain: A service function chain is an ordered or partially ordered set of abstract service functions (SFs) and the ordering constraints that must be applied to packets, frames and/or flows selected as a result of classification. The implied order may not be a linear progression as the architecture allows for SFCs that copy to more than one branch, and also allows for cases where there is flexibility in the order in which service functions need to be applied. The term service chain is often used as shorthand for the service function chain.* For example, in Figure 4, *XYZ.com* is a web service, a *network service* provided by some ASP. It consists of two *service functions:* a *proxy* and a *web server*. A mandated flow from the proxy to the web server is a *service chain*.[9]

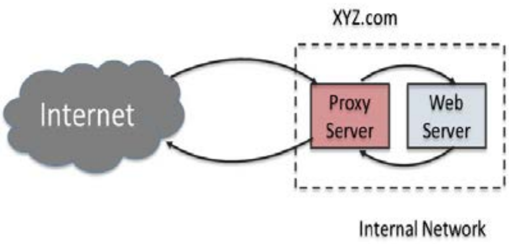

Figure. 4. Network Service, Service Functions and Service Chain.

B. Problem Space:

While developing a scalable, dynamic and automated SFC model, several problems need to be addressed. Some of the major problems identified in the context of SFC framework are as follows [14].

- *Topological Dependencies*: Since most of the service functions are implemented at the hardware level, such as the firewalls, the load-balancers, etc., these service functions are coupled with the underlying physical topology. Service functions need to be added or deleted from the chains and often need to follow a strict ordering. Hence, deletion or addition of these functions gets extremely complicated as it may require changes in the physical topology as well, such as the addition or the deletion of new links and/or traffic engineering. It prevents the network operators from optimally utilizing the network resources.

- *Configuration Complexity*: Dependence on the physical topology leads to the use of static service deployments and static service function chains. Simple actions such as changing the order of the service functions may require changes to the physical topology, which are generally complex and time consuming. This results in slow service provisioning, misconfigurations of the services and sub-optimal utilization of the resources.

- *Flexible Service Delivery*: The dependence on the topology also results in limited flexibility of the service delivery as it may demand significant changes in the current network configurations. Network operators have no consistent way to impose and verify the placement and ordering of the service functions.

SDN is characterized by its three features, which are (1) abstraction of hardware (2) Centralization of policy and control (3) programmability. It is worthy to note that the problems above have been present in networks for a long time. Advances in SDN have already addressed most of these aforementioned problems [27, 58]. Since NFV deals with implementing the network services as software, SDN plays an important role in the orchestration of NFV [58, 65, 110]. However, SFC architecture has some unique features, which mandates these issues to be revisited. This is due to the fact that SFC deals with the resources spread across multiple data centers in geographically distributed areas. The problem of function placement with service chaining, though similar in nature, has significantly different characteristics. For example, SFC is an abstracted view of the ordered service functions, which may or may not be virtual. The order in which the functions need to be visited is defined by the traffic flows dynamically. This is a unique feature of service function chaining architecture and may impose additional constraints such as dynamic traffic steering, ordered flow of services, dynamic path selection and others on the already proposed optimization solutions in the literature[81].

Other issues include consistent ordering of service functions, application of service policies, transport dependence, limited end-to-end service visibility, and deployment of multi-vendor service functions [8, 14]. Now, services may be deployed as a software in a virtualized form, called as virtual network functions (VNFs) and these functions may be moved across the network, without changes in the actual physical topology [13, 40, 96]. In the subsequent sections, we describe the approaches for a systematic SFC architecture and implementations to alleviate the aforementioned problems in the SFC architecture. ITU-T has addressed a similar set of problems while specifying the cloud computing infrastructure requirements [44]. Extending the work of ITU-T, ETSI proposes the use of Network Function Forwarding Graphs (NF-FG) and VNF-FG to provide interconnection and proper packet forwarding among virtual network functions [42, 43]. In the next section, we focus on the architecture proposed by the IETF for the implementation of service function chains, which has been built on top of the NFV architecture proposed by ETSI [40-43].

IV. Service Function Chaining Architecture

In this section, we discuss the architectural approaches proposed towards the implementation and automation of SFC in multi-cloud environments. The aim is to address the problems mentioned in the previous sections, so that service deployment is less complex and less expensive. At the end of the section, we look at the practical implementations of the SFC framework, based on the architectural approaches presented earlier in this section.

A. Service Description and Discovery:

The first step towards implementation of a successful SFC model is to discover and describe the network services. A significant amount of work has been done and is being carried out for service description and standardization, especially for web services. Before we delve into the architectural aspects of SFC, first we need to focus on standard ways to describe and locate the network services. One of the most prominent ways for the service description in the Internet is Web Service Description Language (WSDL) [16, 48]. As mentioned by John et al. [48], WSDL helps define web services from a technical perspective, including the aspects of a service, its interfaces, operations, endpoints, binding, and type definitions. Web service description languages provide a distributed computing infrastructure for both intra and inter-enterprise application integration [17, 48]. However, service function chains can be considered as a particular case of service compositions. WSDL is not sufficient for the specification of network services [48]. For example, current WSDLs may not support dynamic traffic steering among the various branches of the service functions, which may be formed due to SFC. Beek et al. and O'Sullivan have focused on the definition of non-functional properties of the electronic services with the purpose of improving the technology of automated service discovery, comparison, selection and substitution [18, 19]. Based on this work, Cardoso et al. [20] proposed a Unified Service Description Language (USDL) aiming to describe Internet services from business, operational and technical perspectives. USDL has significant applicability to Internet service discovery, comparison, evaluation and management, enabling rigorous decision-making by service requestors [16].

There has been significant work done towards Cloud Services as well. As defined by Sun et al. [16], cloud service declarative definition language (CSDL) [21], SUN [22], and SMI [23] focus on the description of the services in the clouds. Sun et al. [16] provide a detailed description and discussion for these languages. The service description languages (SDLs) developed so far fail to adapt to the contemporary elastic service environments. New data center network and cloud architectures require more flexible SDLs. Web service orchestration has been widely studied already [49, 59] and virtual function chaining problem is similar to it in nature, as both problems tackle the issues of interconnection to multiple instances of the functions or services. However, as stated by Sun et al. [16], there is a lack of a comprehensive specification model for cloud services covering multiple perspectives. The semantic expressivity should also be considered as one of the most important dimensions while developing specification models for cloud services since SFC has additional constraints to be satisfied, such as QoS, SLAs, and policies.

Establishing and maintaining the interconnection among virtual functions is not the only challenge in the SFC architecture. Updating the service chains is even more challenging task. With the recent outburst of mobile apps, network traffic has become extremely dynamic and users expect seamless dynamic changes in the underlying network. SFC architecture should be able to modify the underlying service chains as per the traffic demands. Recent work in the domain of network programming languages (such as Pyretic [24] and Maple [25]) shows how to implement network functionality by controlling the flow space in an OpenFlow switch in a programmatic manner. However, this needs to be adopted to suit the specific requirements of the SFC architecture. In the rest of the section, we discuss advancements towards development of service function chains to enable agile service flow modifications to support elastic service delivery.

B. SFC Architecture:

IETF has taken initiatives towards developing the formal architectures for SFC. The IETF has been building on top of the basic NFV architecture proposed by ETSI. With NFV architectural framework, ETSI has outlined the architecture to support VNF operations across different hypervisors and computing resources. ETSI has proposed a software architecture with VNFs as a building block to construct VNF forwarding graphs. The aim is to redirect the traffic correctly and efficiently in the network. ETSI has also proposed interfacing of the management and the orchestration of NFV with other management systems such as Element Management System (EMS) or Network Management System (NMS). NFV Infrastructure (NFVI) is central to the framework which supports the execution of VNFs [42, 43].

Figure 5 shows how end-to-end communication happens in a network service, where Point A is a start point and Point B is an end point. This is achieved using a VNF forwarding graph (VNFFG), or simply NFF, as shown in Figure 5. The end points are connected to the network functions via the available network interfaces. Virtualized network functions run on top of the underlying hardware resources. Such intermediate forwarding elements, called as Network Function Forwarders (NFF), are necessary as the network traffic needs to be steered dynamically. NFF acts like a building block for the VNF-FG. A VNF (or simply NF) may be composed of one or multiple components and appear as a single box from outside the system.

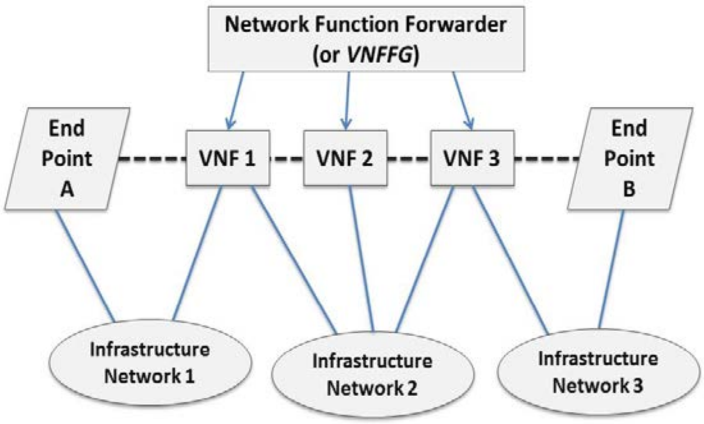

Figure. 5. End-to-End communication using VNF Forwarding Graphs (VNFFG).

The SFC architectural approach proposed by IETF (Quinn and Elzur) [7] suggests implementation of the data-plane for carrying information along the service path. Use of Network Service Header (NSH) has been proposed for the same. NSH may be considered as an enabler of the NFF concept. NSH contains metadata and service path information that are added to a packet or frame and used to create a service plane. The packets with the NSH are then encapsulated in an outer header for transport. The service header is added by a service classification function – a device or an application – that determines which packets require servicing, and correspondingly which service path to follow to apply the appropriate service. Implementation of NSH provides a mechanism to carry the information flow from one forwarding element to the next, which is important to steer the traffic along the correct path. The NSH-aware nodes may add, remove or update these headers. Generally the first node in the chain adds these headers and last node in the path removes these header fields.

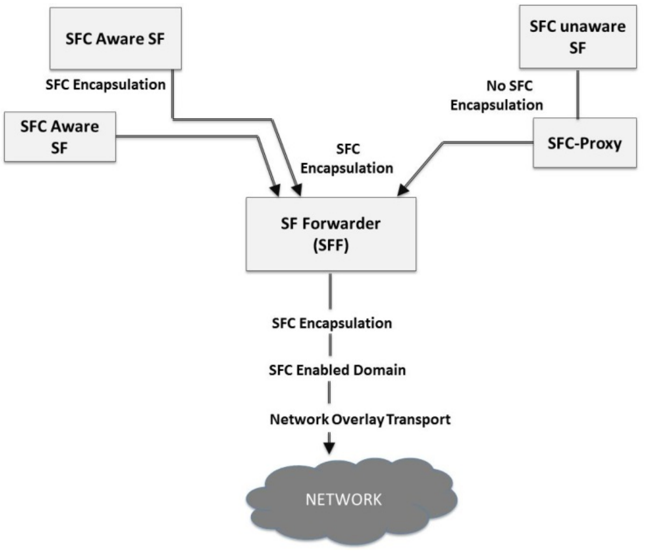

Figure. 6. System Function Chain Architecture and Components.

Base header provides the service chain path information and intermediate nodes called as "Service Function Forwarder" (SFF) might use it to determine the service path information (Figure 6). SFF is a node in the service function chain, which implements NFF. These headers can be used to embed the service policy selection information. For example, a "context" field in the header can be used to embed the local policy related information. Proxy nodes can be used to accommodate NSH-unaware nodes in the network for backward compatibility. Thus, NSH helps to achieve topology independence with metadata sharing. Detailed description of header fields is out of scope of this article; however, readers may refer the draft [7] for more details.

C. Service Function Paths:

At an abstract level, the service function chain is an abstracted view of a service that specifies the set of required SFs as well as the order in which they must be executed. Service function chains may start from the origin of the service function graph (i.e., node 1 in Figure 7), or from any subsequent node in the graph. SFs may, therefore, become branching nodes in the graph, with those SFs selecting edges that move traffic to one or more branches, depending on user service demands. Service function chains may have more than one termination points [8] as well, as shown in Figure 7. A service function path (SFP) is a mechanism used by service chaining platform to express the result of applying more granular policy and operational constraints to the abstract requirements of a service function chain [7]. The SFC architecture introduced by Halpern and Pignataro proposes the use of network function forwarder (NFF) to determine SFPs dynamically and enforce the policies at run time (Figure 7) [8]. NFF performs similar tasks as VNF-FG proposed by ETSI as defined earlier in [8, 70].

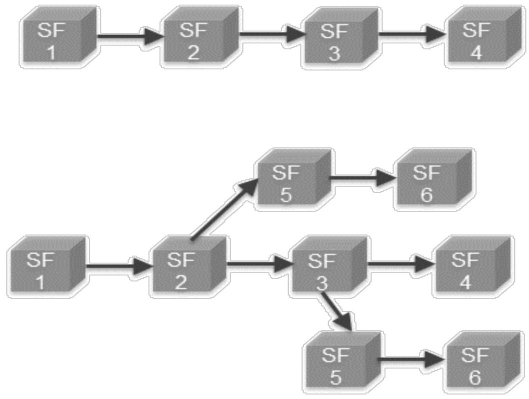

Figure. 7. Network Service, Service Functions and Service Chain.

Further, the architecture allows two or more SFs to be attached to the same SFF, and possibly connected via internal means allowing more effective communication. A SFC control plane may be implemented to provide an SFC-enabled domain wide view of all available service function resources as well as the network locators through which they are reachable. Also, the SFC control plane will provide requisite SFC data-plane information for the SFC architecture components [7, 8]. In the next sub-section, we describe various implementation efforts towards a feasible and practical SFC platform. We observe that these proposed designs implement the concepts of SFFs for dynamic traffic steering and provide some customizations.

D. Implementation Perspective:

The SFC model needs to be adapted to fit in the carrier-grade telecommunication networks. There is a need for a systematic way to study and implement the SFC platform for carrier networks. It is important to have a quick look at the practical implementations of SFC which are available in the literature, so that the practicability of the architectures proposed so far may be tested. Scalable and dynamic traffic steering capabilities proposed in the practical implementation of the SFC model could help the middle-box deployment and establishment of SFC platform for the network operators [105-108]. Below we discuss some of the approaches for the development of the SFC model, especially using the advancements in the field of SDN [2, 5].

To be able to steer traffic based on the preconfigured service policies and placement of functions, there is a need for traffic steering policies. Quinn and Nadeau, have proposed baselines in IETF drafts [14]. Going a step further, Zhang et al. [6] demonstrate the need for steering traffic at the granularity of subscriber and traffic types. The authors propose an approach "SDN inline services and forwarding (StEERING)" based on SDN. The focus is mostly to steer the traffic among subscribers, network services and the Internet. Service chaining problem is addressed in this work by reusing OpenFlow multiple tables [26].

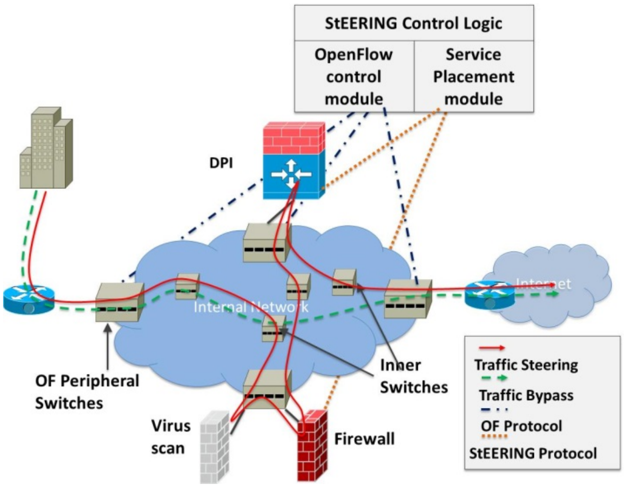

Figure. 8. System Architecture of StEERING.

The concept is that of placing OpenFlow switches on the periphery of the service delivery network and configuring these switches for the various flows using an OpenFlow controller. These switches are expected to classify incoming traffic and steer them to the appropriate service. StEERING architecture is shown in Figure 8. StEERING consists of two modules, a standard OpenFlow controller and a logic controller which periodically runs to determine the best location for the inline services. OpenFlow controller classifies and forwards the incoming flows to the proper channel. Policies are determined using the packet headers or packet payloads such as a URL. This is a novel and practical approach but a limited one. For example, the issue of dynamic resource allocation has not been addressed in the design, which is equally important as dynamic traffic steering.

Scalability in StEERING is achieved by avoiding the exponential growth of the forwarding rules with the network flows [6]. This is achieved with the help of specific design choices to reduce the states to be maintained at each switch such as using multiple tables and introducing new meta-data types. In addition, a greedy heuristic is implemented for the placement of the inline services to minimize the delays for the end users. Authors validate the prototype with the measurements from a lab setup and a simulation model. Al-Fares [33] presents a scalable network architecture for the datacenters, especially from the multi-cloud perspective.

Blendin et al. [13] propose mapping packets to service chains at the edges and forwarding them to service instances using OpenFlow. Use of IP addressing at ingress and egress routers is proposed for proper mapping at the edges, however, forwarding inside the OpenFlow network is strictly layer-2 based. The high level architecture of the proposed solution is depicted in Figure 9. The concepts of service function chaining controller (SFCC) and the service function chaining router (SFCR) have been introduced in this work to separate the functionality in various layers. Design approach for SFC proposed in [13] addresses on-demand policy change and path selection based on the policy.

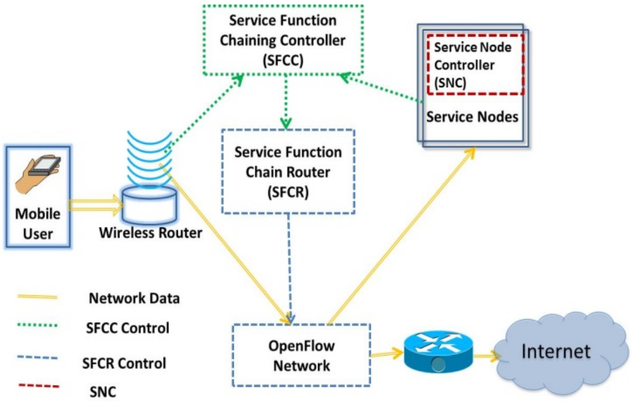

Figure. 9. System Architecture proposed by Blendin et al.

Quinn and Guichard propose an architecture based network service headers to construct topological independent service paths needed for the end-to-end service function chains [82]. However, besides the SFC implementation itself, placement of the network services and virtual function in the given network are equally important to improve the network resource utilization and the end-to-end delay to the end users. Another important problem related to SFC is the resource allocation. In the next section, we discuss some of the important research work, especially from the optimization perspective. These techniques need to be studied in detail and tailored to address issues of the SFC architecture mentioned earlier.

V. SFC: Optimization Approaches

Various solutions discussed so far towards the standardization and implementation of SFC have addressed the unique and unprecedented challenges imposed by the new Service Chaining architecture. However, they lack analytical models and performance analysis techniques for the proposed solutions. Besides the practical implementation of NFV and SFC, emphasis on optimal and automatic placement of the network services and mapping of these services to the available resources over the underlying physical or virtualized hardware is equally important. Placement of virtual functions and network services in the given network to minimize end-to-end latency for end users is becoming an important problem, especially considering QoS and SLA constraints.

By addressing various optimization models below, we make an attempt to address the different challenges in the SFC architecture, discussed in Section III-B. These problems have been identified in [10] as major challenges for practical implementation of SFC. Such models address important problems in the multi-cloud environment, such as cost, energy consumption, network traffic volume, latency and many others. An important point to note is that the following optimization studies, though not directly applicable to the SFC architecture, can be applied to service function chaining (SFC) by addressing the extra constraints imposed by SFC model.

A. Network Latency Minimization:

Latency is an important factor in measuring the network performance. Total latency in the network needs to be minimized, in any kind of network, including SFC architecture. Different users can tolerate various amounts of delays and ASPs should be aware of the users' categorization. Multiple instances of the VNFs as per users' locations and their concentration is one way to minimize the latencies in the network. A set of heuristics is presented in [61] to present a constrained mirror placement approach in Content Distribution Networks (CDNs) to minimize the latencies. The authors demonstrate that the results can be obtained for the resource allocation problem within acceptable time limits in a real time scenario with little compromise in the solution quality. The focus of the proposed optimization model is primarily on the network latency. Additional algorithmic approaches such as "Min K-Center" and "l-Greedy" are used in addition to the sub-modular set cover to reduce the computational complexities and the execution time with little compromise in the solution quality.

An approach for reducing control traffic in managed network has been proposed in [11] to reduce the total network latency. The authors propose to divide the larger networks into smaller sub-networks to limit the broadcast domains of the control traffic leading to reduction in latency. However, the integer linear programming (ILP) model needs to be modified to cater to the needs of SFC architecture. A primitive study of response time optimization for function chaining has been done in [62], however, similar research in other areas of optimization in SFC, needs to be carried out.

A server selection scheme based on the optimal routing is another approach to reduce the response time or total latency [3]. A trace-driven simulation method is used in [3] to demonstrate the results and improvements by the proposed scheme. The work relies on the complete path from source to destination incorporating the network delays rather than just considering the shortest distance as a selection criteria. The authors try to minimize the cost function, that is total delay D_{ij} between nodes i and j (given below), where C_{ij} is the link capacity, d_{ij} is the propagation and processing delay and F_{ij} is the amount of data on the link. The problem considered is an optimal routing problem for Content Distribution Network (CDN) architecture. Authors present simulation results, especially, to measure the response time of HTTP requests. For this purpose, they have considered WorldCup98 logs for a specific day and requests were collected across four different servers in USA and Europe [3].

$$D_{ij} = \frac{F_{ij}}{C_{ij} - F_{ij}} + d_{ij} \times F_{ij} \qquad (1)$$

The problem of location of the VFs and allocation of the resources to VFs in the service function chain is, though similar in nature, inherently different in principle. For example, in a service function chain, one needs to consider not only the identification of controller locations and allocation scheme but also the interconnection among the network service instances and its impact on service level agreements (SLAs). Also, parameters such as datacenter capabilities and inline service demands need to be considered as well, which may affect the network service distribution in the clouds [87].

B. Resource Utilization Optimization:

Since cloud resources are expensive, resource optimization in the network has always been a major concern in the research, and SFC framework is no exception. User demands and user locations in the contemporary networks are highly flexible. This is due to the recent outburst in the usage of mobile devices [60]. As a result, frequent changes to the deployment of the resources need to be done. Dynamic resource allocation of the VMs and eventually of the network services and virtual functions is another problem of significant importance. Various resource allocation algorithms have been proposed in the literature so far [76]. Dynamic resource allocation to perform task scheduling of virtual machines on the servers in data centers has been proposed in [10]. The authors initially discuss major design challenges for network aware resource allocation system. Such challenges are outlined in Table II. Finally authors provide scheduling technique to schedule VMs on processors to minimize total communication cost. The focus of the work has been on the network aware resource allocation, and cost as well as power minimization.

TABLE II. CHALLENGES IN NETWORK AWARE RA SYSTEM

Data Locality [30, 62, 97]	Providing data-aware scheduler
	Data Intensive Apps
	Move Data vs. Move Apps
Internal DC Network Reliability [33, 36]	Design of a resilient virtual network
	Complexity Vs. efficiency
	Enforcing SLAs
SDN Design Challenges [25, 27, 32, 54, 76]	Reliability
	Scalability
	Visibility
	Controller placement problem

Under-utilization of the network resources is a frequent problem in the dynamic resource allocation case due to frequent changes to the deployed resources and ad-hoc deployment of the resources by network operators. Scheduling schemes are presented in [85, 86] for the placement of network functions so that the resource utilization can be optimized. Dynamic resource allocation problem in cloud computing to optimize utilization of network resources and lengthy response times has been considered in [37, 78]. A distributed multiple-criteria decision analysis approach is proposed, where node agents, which are tightly coupled with the physical machines, carry out the configuration task.

Every node agent observes the local resource usage of the physical machine (PM) and performs the configuration step, which is carried out in three steps as VM placement, VM selection and VM migration. The process of dynamically allocating VMs to PMs and maintaining resource distribution among VMs to meet their resource requirements is modeled as a distributed process carried out by individual node agents in the system. The three-step approach helps to reduce underutilization of the network resources and the distributed approach of the algorithms helps in achieving the scalability [37].

A design, implementation and evaluation of resource management systems based on OpenStack [31], an open-source cloud platform, has been presented in [39]. Authors demonstrate the applications of the proposed heuristic from practical implementation perspective. The proposed solution has been divided into two major components, that is, *Initial Placement Controller* and *Dynamic Placement Controller*. OpenStack least-cost scheduler is used in the initial phase [31]. The dynamic placement controller continuously adapts the placements of the virtual machines through live-migration. Though the proposed solution has been demonstrated in practical environments, the selected approach of VM migration may prove resource intensive. Authors in [84] propose a model for efficient virtual function placement and forming optimal service chains to reduce overall resource utilization. Various NFV aware optimal resource allocation approaches have been proposed in [100-104]. For example, Basta et al. apply SDN for the service function placement problem using NFV in LTE mobile networks [104]. Authors in [101] optimize the total traffic for optimal network resource utilization. Such approaches are necessary to reduce the overall cost and energy consumption in NFV scenario.

C. Cost Minimization:

SFC architecture is a replication based architecture where multiple replicas of service functions need to be deployed across the network. Selecting servers or host machines to deploy the service functions is an important problem as far as limited infrastructure resources are considered. To begin with, the most important question to be answered is what kind of and how many of the virtual machines are to be installed, as well as, at what locations these VMs need to be deployed to satisfy all the user demands so that the total reservation cost is minimized. The problem has been proven to be NP-complete [102]. An approximation algorithm over sub-modular set cover (SSC), with its approximation bounds has been presented in [12, 80] and its performance has been demonstrated close to optimal. The proposed algorithm can be used by both enterprise and cloud service providers in real time environment to reduce costs or for a higher degree of resource sharing. Resource requirements are considered in terms of memory, computational power and disk space. With these simplified configurations, resource requirements can easily be mapped to the nearest available configuration and the complexity associated with heterogeneity of the resource is minimized [102].

A similar problem of optimizing resource allocation for multimedia cloud has been studied in [30] to reduce the overall cost. Using a queuing model, a priority service scheme in a multimedia cloud datacenter has been modeled. Three concatenated queuing systems are proposed by the authors: (1) schedule queue, (2) computation queue and (3) transmission queue. The optimization problem has been divided into two sub-problems which are resource-cost minimization problem and service-response time minimization problem. In the first part, the authors try to minimize the total resource cost by jointly optimizing the scheduling rate at the master server, the computing rate at the computing server, and the transmission rate at the transmission server, subject to the stability constraint in each queuing system. The authors also consider the service response time constraints for each class of service. The problem has been proven to be a convex optimization problem and use of primal-dual interior point method has been proposed to reach a solution. In the later part, the authors try to minimize the total response time for the end users with multimedia traffic [30].

D. Power/Energy Minimization:

Studies have shown that the energy is one of the most significant factors in the OPEX of datacenters and clouds. For cloud providers, it is important to reduce the energy consumption to reduce costs as well as improve reliability of networks [55, 56]. Hence it is important to have a look at the options for the energy minimization in the SFC framework. We have a quick look at the important research work done for energy minimization in multi-cloud environments, which may be applied to the SFC framework.

To reduce power consumption in the clouds, power-aware provisioning of virtual machines for real-time cloud services has been proposed in [56]. Proposed Adaptive-Dynamic Voltage Scaling (DVS) and δ-Advanced-DVS schemes demonstrate profits with reduced power consumption. When a datacenter receives a request from resource broker, it returns the price for service-provisioning if it can be provisioned at all. The broker selects the datacenter with minimum provisioning cost. A threshold based approach has been proposed in [55] to allocate the resources in heterogeneous cloud systems. A mathematical analysis has been provided to determine the threshold value. A power aware algorithm has been proposed to make decisions on ordered server lists, server activation thresholds and workload distributions for resource allocation.

A need for an energy efficient mechanism for clouds, has been demonstrated in [57]. A system called "*Jitter*" has been developed for Message Passing Interface (MPI) standards. MPI is a specification to be used by the developers and users for message passing libraries. Jitter exploits slack time spent by nodes at synchronization points by reducing the energy consumption on those nodes, which in turn significantly reduces the consumed energy. The authors demonstrate 8% energy reduction with as little as 2% execution time penalty. Such mechanisms need to be investigated further in the context of SFC architecture to reduce energy consumption.

E. SLA Based Optimization:

Provisioning network services meeting SLAs in the network is an important factor to be considered by network administrators. An SLA-based optimal resource allocation for multi-tier service model in multi-cloud environment has been proposed [34]. Figure 10 explains the multi-tier service model. The sample client request shown in the figure demands an ordered set of three tiers of the application. At each level of application tier, the client request is distributed among the appropriate servers. Forward and backward requests, represented by solid and dashed lines respectively, are served by different queues in order to simplify the queuing model. In Figure 10, the triangles represent the forwarding elements and rectangles represent the queuing elements, while the arrows represent the direction of data flow. The authors then derive the equation for the average response time based on the model. Additional constraints are added to accommodate different SLAs. The authors also present a model to optimize the profit from the SLA contracts and loss from operational costs.

Going further, Dib, Parlavantzas and Morin [35] consider Platform as a Service (PaaS) in a multi-cloud environment and provide an SLA-based profit optimization model. On the contrary, Pires and Baran [36] focus on virtual machine placement problem considering SLA constraints, which holds the significance from the practical implementation perspective. Such options may enable network service providers to maximize profits and invest towards proliferation of SFC models in the future. The work in [52] considers Infrastructure as a Service (IaaS) for enterprise networks to meet the SLA requirements. Total profit to the service providers and the satisfaction to the users depend on how the system meets the SLA agreements. For the success of NFV and SFC technology, it is imperative that various SLA related constraints are satisfied optimally.

F. QoS based Optimization:

One cannot ignore the importance of QoS in the networks while studying the NFV architecture problem [9]. Service providers should take QoS requirements into consideration to meet the promised SLAs for demanding applications. A dynamic resource allocation approach in clouds while considering Quality of Service (QoS) has been proposed in [38]. Processing (CPU) and networking abilities of the servers have been taken into consideration while formulating the model. The authors propose enhancements to the center selection algorithm [32] for resource allocation to consider other heterogeneous resource attributes as well as QoS.

Datacenter clusters are classified based on the level of the QoS that can be provisioned with the available resources. A datacenter that can provide multiple resources required by the request is selected and resources are allocated accordingly. Once the service time has elapsed, the resources are released and added to the pool again. Using simulations, the authors have demonstrated a 30% reduction in the required resources with the suggested algorithm compared to the standard center selection algorithm. A mechanism to allocate virtualized resources to the VMs with stochastic optimization model to analyze the performance with QoS has been proposed in [53]. Authors have considered non-cooperative environment with VMs being selfish, that is, VMs trying to maximize their own profits. Authors have proved the method to be optimal theoretically.

G. Other Approaches:

There are other approaches for optimal resource allocation, such as, auction based, gossip-based, and broker-based. We have a quick look at some of them in this sub-section for resource allocation in clouds, which may be applied to the SFC problem as well. An auction based approach for resource allocation in the clouds has been proposed in [50]. In the proposed solution, CSPs collect the users' bids and the resource distribution is done among *K*-highest bidders. As explained earlier, a broker-based approach is presented in [55] to choose virtual machines for allocation. A distributed mechanism has been developed in the work to allocate resources in large cloud environments. The authors demonstrate that their results are close to optimal. Also, a gossip based approach for resource allocation in multi-cloud environments has been proposed in [54], where requests for the resources and replies from the resource providers are forwarded in the network with some probability, which are derived dynamically.

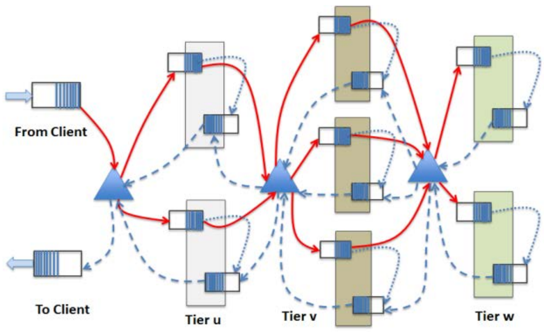

Figure. 10. Multi-tier (three tier) service model.

An automated solution for resource allocation in the multi-cloud environments has been presented in [29]. The focus has been on the alternative approach to the auction based solutions where providers and consumers automatically negotiate the resource leasing contracts. The authors demonstrate that auction based approaches [47] does not necessarily result in the most efficient solution. Hence, a negotiation based approach is suggested so that multiple providers may pitch in to satisfy consumers' resource requirements. The authors initially discuss standard negotiation protocols and then propose modified "buyer's negotiation strategy" and "seller's negotiation strategy" for improved performance.

Kuribayashi [32] proposes a multiple resource allocation method for clouds. The model takes fair joint multiple resource allocation into consideration, which may be applied to NFV optimization problems. The four principles proposed by the author to obtain fairness in joint multiple resource allocation are given as: (1) Non-delay resource allocation, that is, fairness without queuing, (2) Fair allocation of multiple resources: Unlike the other works, which primarily focus on a single type of resource, in this work, the author focuses on the allocation of multiple types of resources, which is important in contemporary networks. (3) Equalizing the total amount of processing ability and bandwidth for all users would not achieve fairness and (4) Fairness with no service request rejection occurs for any user even if the amount of resources allocated are not balanced. A normalized value of fairness is calculated for the final evaluation. This helps the author achieve joint multiple resource allocation.

Various models and architectures have been discussed above addressing a wide range of issues related to SFC and its role as an enabler of the NFV architecture. We observe that these approaches lack analytical study for the end-to-end service model for SFC and, hence, there cannot be a comparison between optimal solutions and the heuristic approaches for SFC models. To form an analytical model for SFC to study its behavior, the related optimization problems need to be solved considering extra parameters and constraints imposed by the SFC architecture (such as queuing delays, policies and others). We now discuss the open research problems which need to be addressed for the successful and efficient deployment of the SFC architecture.

VI. Open Issues and Challenges

A comparison of the research work presented in this article is given in Table III. Rows of the table represent the reference for the research work under consideration and columns represent the research area covered in that particular research work, which is related to the SFC problem. The table demonstrates limited research in different areas related to SFC. It must be noted that many issues, such as security, dynamic load balancing, application of SLAs, high availability, QoS and others need to be studied in depth in the context of SFC. The parameters which need to be covered for SFC implementation include dynamic traffic steering, dynamic resource allocation and the end-to-end service delivery. We observe that, though a significant amount of research has been done in related research topics, major issues are still open from an SFC perspective. A significant amount of work needs to be done to establish a stable and reliable architecture for SFC. Important areas which are still open for research are discussed below:

A. Optimal Resource Allocation:

As mentioned in earlier sections, optimal resource allocation is a widely studied topic; however, it is still an open issue for SFC architecture. This is due to the fact that SFC deals with the resources spread across multiple clouds in geographically distributed areas. This condition imposes additional constraints on the already proposed resource allocation strategies, which mostly consider single-cloud scenario. Current research studies of resource allocation in virtual and cloud environment can be applied to service chaining problem, but not without considering additional constraints of the SFC architecture [71]. As outlined in Section V, a significant amount of work has been already done in the field of virtual function placement as well as resource allocation [97-99, 120]. Ghaznavi et al. have considered elastic virtual functions placement problem [121]. The problem of function placement with service chaining, though similar in nature, has significantly different characteristics. For example, the service function chaining problem imposes additional constraints such as an ordered chain of functions. Considering such additional constraints, a modified optimization model needs to be developed to suit the SFC framework.

B. Dynamic Traffic Steering:

Traffic steering is an important aspect of SFC model. The steering has to be dynamic. For example, the path followed by the user request may be different depending on what is the target functionality, even though the request packet is generated by the same user. In addition to the dynamic routing, the routing decisions taken at the SFF modules have to be efficient. Implementation of SFF has been proposed in the standards to handle the dynamic routing in service function chains, however, their design and architecture is still an open issue of research.

C. Dynamic Service Mapping:

Dynamic nature of service function chaining imposes an additional set of challenges. Zen, Song and Chen [28] have addressed the problem of dynamic resource allocation in the clouds as discussed earlier. However, the proposed solutions do not address identification and handling of packets if the packet header information is updated, leaving the issue of mapping among users and service chain instances based on the required resources still open. Qazi et al. discuss the challenges in depth in their work [103].

D. Enforcing Service Policies:

Researchers have become aware about the importance of service policies and their enforcements in the multi-cloud environments. As outlined earlier, a significant amount of work has been done already for SLA enforcements in virtualized and dynamic multi-cloud environments. However, the problem has to be revisited in the context of SFC models. SFC has different paths (or chains) which need to be followed by the packets. It may be a case that each path is dealing with different ISPs and/or CSPs and may have a different set of SLAs which need to be satisfied. Policies may dictate connecting a group of subscribers to a particular point of presence or routing traffic through firewall and intrusion detection function. There may be a policy to route all video streams through the Broadband Remote Access Server (BRAS) to record per flow information for billing purposes. Each policy may be implemented as one or more service chains [111]. A service chain would in turn consist of an ordered set of one or more VNFs that process a particular class of traffic in the tenant requested sequence. Such additional constraint of dynamic SLA mapping may need significant rework to the existing models to be effective. Also, the SFC architecture has to implement the QoS and it may not be trivial as such constraints are getting stringent.

E. Security Issues:

Security is another major concern for SFC architecture. There have been recent studies towards the security in cloud and virtualized environments [113-115]. However, a newly proposed NFV and SFC framework may be vulnerable to new pitfalls which may be exploited by the attackers. For example, with the advents in the distributed Denial of Service attacks (DDoS) [46], attackers may target individual SFF, increasing the probability of success. Failure in single SFF means failure of a complete service chain. IPSec and TLS [116-118] mechanisms which are deployed to protect the links between two communication entities in SDN, may be used to protect the new interfaces introduced by NFV scenarios. However, these mechanisms need to be automated to suit the needs of NFV and SFC infrastructure. Researches have started to explore the security threats and solution if NFV domain, for example the work proposed by Battula [112]. Different research options such as SFF redundancy, anomaly detection at SFFs needs to be explored and still open for research.

TABLE III. LIMITED RESEARCH WORK RELATED TO SFC

Referred material	Practical Implementation	Optimization							Security	Scalability	SFC Standards /Surveys	NFV Standards /Surveys	SDL and Service Measurement	Gossip/Broker/ Auction Based	5-G Networks
		Traffic	Energy	Response Time	Cost	Resource Allocation	VM Placement	SLA/QoS							
[2], [6], [13], [25], [60], [82], [90], [95]	✓									✓					
[97], [98], [99], [100]							✓								
[7], [8], [9], [48]											✓				
[10], [28], [37], [77], [78], [84], [85], [86]						✓									
[75]	✓														✓
[3], [11]		✓		✓											
[12], [30],[80], [102]					✓	✓									
[101], [103]						✓	✓			✓					
[5], [14], [40], [41], [42], [43], [49], [96], [100]												✓			
[16], [17], [18], [19], [20], [23]													✓		
[32], [76], [95]	✓					✓									
[34], [35], [52], [53]								✓							
[36]							✓	✓							
[38]						✓		✓							
[39]	✓					✓									
[47], [50]														✓	
[56], [57], [58], [59]			✓												
[62]					✓	✓	✓								
[29], [54]						✓								✓	
[55]			✓											✓	
[77], [79], [81]												✓			
[104]							✓					✓			
[105], [106], [107]	✓											✓			

F. Reliability and Service Availability:

The services offered by ASP as a set of functions should be available all the time for the end users. A dynamic addition or removal of a function of the service needs to be done seamlessly without interrupting current services. Dynamic methods should be made available to check the service chain connectivity. The methods such as those proposed in [63] are suitable for static models and needs to be adopted to suit the dynamic nature of the SFC model. There is a need to develop mechanisms for fault detection and fault isolation from the NFV and SFC perspective.

G. Interoperability:

Service function chaining is expected to play an important role in the deployment of Internet of Things (IoT) networking paradigm [74, 79]. However, IoT imposes significant challenges such as a large number of heterogeneous sensory devices willing to communicate with each other [78, 88]. SFC architecture should be able to provide interoperability among such heterogeneous devices for successful end-to-end communication and success of IoT deployments.

H. Deployment:

With the advent of NFV and SFC, the problem of deployment of such dynamic function chains has gained a significant traction in the literature. For example, the authors provide a scheme for online admission control and embedding of service chains in [4, 122]. The authors try to address the problem of how to optimally admit and embed service chain requests over the available infrastructure and resources. On the contrary, a polynomial time approximation algorithm considering an offline batch embedding of multiple service channels is proposed in [51]. The authors consider the objectives of maximizing the profit by embedding an optimal subset of requests or minimizing the costs when all requests need to be embedded. Lukovszki et al. propose deterministic-greedy approximation algorithms for incremental middle-box deployment so that distance constraints between communicating node pairs as well as capacity constraints on the network nodes are satisfied. Kuo et al. consider the problem of deploying service function chains considering links and servers usages [123]. However, as pointed out earlier, many more challenges are still to be addressed, as far as efficient deployment of service function chains is considered, such as the deployment of SFCs in heterogeneous environments considering the SLAs and QoS. Further research is needed for efficient deployment of virtual functions, especially in the context of IoT.

I. Monitoring, Management and Orchestration:

A significant amount of work has been done by ETSI for the monitoring and orchestration of NFV. As a result, NFV MANO framework has been developed [105]. The proposed VNF manager (VNFM) works in concert with other NFV-MANO functional blocks, such as the virtualized infrastructure manager (VIM) to help standardize the functions of virtual networking and increase the interoperability of software-defined networking elements. The authors in [99] also propose an optimization model for orchestration of NFV. Though the research works have proposed some schemes for monitoring, management, and orchestration in the NFV and SFC context, a detailed study is still missing. A comprehensive end-to-end management platform is mandatory for better operations, administration and management (OA&M) of service function chains.

VII. Conclusions

With the growing demand of clouds and the cloud services, SFC and NFV are gaining popularity among the application service providers, internet service providers and cloud service provides. NFV is proving to be an effective and flexible alternative for the service deployments across multiple-clouds. SFC is the technology which makes NFV feasible. Hence, it becomes important to study the current state of the art for SFC, from theoretical as well as practical perspectives.

Emergence of new networking paradigm such as IoT has led to a recent spurt in interest in service function chaining and network function virtualization. In this work, first we have analyzed the research directions towards the implementation of SFC in the contemporary networks. Then, we have discussed the scope of SFC, especially various research and standardization works being carried out in organizations such as ETSI, ITU-T and IETF SFC working group and IRTF NFV research group. We have glanced at various research works which have tried to identify the SFC issues and have proposed the enhancements which need to be incorporated to be applicable to dynamic service chain creation. First, we have discussed the SFC problem space and then research directions such as service description languages, SFC data-plane and other possible architectures to define the service flows and the service function forwarders.

Furthermore, we discussed the possible approaches towards the implementation of service function chains in the contemporary networks such as traffic steering. We also glanced at the optimal resource and the dynamic resource allocation approaches as well as the virtual function placement problem to implement an efficient SFC model. We have demonstrated the need for relevant modifications in the contemporary research works to make them suitable for the SFC architecture considering the additional constraints imposed by the SFC model.

Research topics which are still open include optimization strategies for decomposition and aggregation of service functions, service modeling and description languages, design for mobility and scalability, dynamic chain formation and modification, dynamic and agile traffic steering, context awareness and adaptation, enforcement of different service policies, security of service chains and many others. We observe that though a significant amount of work has been done and some novel architectures have been implemented, there is a dearth of in-depth quantitative results for the evaluation purpose. There is a need for a scientific study of the problem of placement of network services and virtual resource allocation, so that users can benefit from minimum latencies and carrier-grade implementations of SFC in multi-cloud infrastructures and operators can benefit from low cost and efficient networks.

ACKNOWLEDGEMENT

This publication was made possible by the NPRP award [NPRP 6 - 901 - 2 - 370] from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the author[s].

REFERENCES

[1] S. Paul, R. Jain, M. Samaka, J. Pan, "Application Delivery in Multi-Cloud Environments using Software Defined Networking," Computer Networks Special Issue on cloud networking and communications, February 2014, pp. 166-186.

[2] S. Paul and R. Jain, "OpenADN: Mobile Apps on Global Clouds Using OpenFlow and Software Defined Networking," 1st Int'l. Wksp. Management and Security Technologies for Cloud Computing, December 2012, pp. 719-723.

[3] S. Bakiras, "Approximate Server Selection Algorithms in Content Distribution Networks," IEEE International Conference on Communications (ICC), vol. 3, May 2005, pp. 1490-1494.

[4] T. Lukovszki, M. Rost, and S. Schmid, "It's a match!: Near-optimal and incremental middlebox deployment," SIGCOMM Comput. Commun. Rev., vol. 46, no. 1, Jan. 2016, pp. 30-36..

[5] M. Chiosi, "Network Functions Virtualisation (NFV)," European Telecommunications Standards Institute (ETSI), October 2013.

[6] Y. Zhang, N. Beheshti, L. Beliveau, "StEERING: A Software-defined Networking for Inline Service Chaining," 21st IEEE International Conference on Network Protocols (ICNP), pp. 1-10.

[7] P. Quinn et. al, "Network Service Header," Internet-Draft, draft-ietf-sfc-nsh-01.txt, IETF, July 2015, p. 43.

[8] J. Halpern, C. Pignataro, "Service Function Chaining (SFC) Architecture," RFC 7665, p. 32.

[9] S. Kumar, M. Tufail, S. Majee, C. Captari, S. Homma, "Service Function Chaining Use Cases In Data Centers," Internet-Draft, draft-ietf-sfc-dc-use-cases-03, July 2015, p.23.

[10] M. Sharkh, M. Jammal, A. Shami, A. Ouda, "Resource allocation in a network-based cloud computing environment: design challenges," IEEE Communications Magazine, vol. 51 no.11, pp. 46-52, November 2013.

[11] D. Bhamare, M. Krishnamoorthy, A. Gumaste, "Models and algorithms for centralized control planes to optimize control traffic overhead," Computer Communications, 70 (2015), doi:10.1016/j.comcom.2015.06.022, pp. 68-78.

[12] F. Chang, J. Ren, R. Viswanathan, "Optimal Resource Allocation in Clouds,," IEEE 3rd International Conference on Cloud Computing (CLOUD), July 2010, pp. 418-425.

[13] J. Blendin, J. Ruckert, N. Leymann, G. Schyguda, D. Hausheer, "Position Paper: Software-Defined Network Service Chaining," Third European Workshop on Software Defined Networks (EWSDN), September 2014, pp. 109-114.

[14] P. Quinn and T. Nadeau, "Problem Statement for Service Function Chaining," IETF RFC 7498, April 2015.

[15] M. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, "Service-oriented computing: State of the art and research challenges," Computer Journal, vol. 40 no. 11, November 2007, pp. 38-45.

[16] L. Sun, H. Dong, J. Ashraf, "Survey of Service Description Languages and Their Issues in Cloud Computing," Eighth International Conference on Semantics, Knowledge and Grids (SKG), October 2012, pp.128-135.

[17] R. Chinnici, J. Moreau, A. Ryman, and S. Weerawarana, "Webservices description language (wsdl) version 2.0 part 1: Core language," Online: http://www.w3.org/TR/wsdl20/

[18] M. H. ter Beek, A. Bucchiarone, S. Gnesi, "Web service composition approaches: From industrial standards to formal methods," Second International Conference on Internet and Web Applications and Services (ICIW), 2007, pp. 15-20.

[19] J. O'Sullivan, "Towards a precise understanding of service properties," QUT Thesis (PhD), Faculty of Information Technology, Queensland University of Technology, 2006.

[20] J. Cardoso, A. Barros, N. May, U. Kylau, "Towards a unified service description language for the internet of services: Requirements and first developments," in Proc. SCC 2010, pp. a602-609.

[21] K. Jussin, K. Waldemar, "Knowledge service engineering handbook," CRC Press, 2012.

[22] Y. Sun, T. Harmer, A. Stewart, P.Wright, "Mapping application requirements to cloud resources," in Proc. Euro-Par'11, 2011, pp. 104-112.

[23] CloudCommons, "Introducing the Service Measurement Index," Cloud Service Measurement Initiative Consortium, 2012.

[24] C. Monsanto, J. Reich, N. Foster, J. Rexford, D. Walker, "Composing software-defined networks," in Proc., 10th USENIX Symposium on Networked Systems Design and Implementation Proceedings of NSDI, 2013, pp. 1-13.

[25] A. Voellmy, J. Wang, R. Yang, B. Ford, P. Hudak, "Maple: simplifying SDN programming using algorithmic policies," Proceedings of the ACM SIGCOMM conference on SIGCOMM, August 2013.

[26] Online: https://www.opennetworking.org/sdn-resources/openflow

[27] D. Kreutz, F. Ramos, P. Verissimo, C. Rothenberg, S. Azodolmolky, S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol.103, no.1, January 2015, pp.14-76.

[28] X. Zen, W. Song, Q. Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," IEEE Transactions on Parallel and Distributed Systems,, vol.24 no.6, June 2013, pp. 1107-1117.

[29] B. An, V. Lesser, D. Irwin, M. Zink, "Automated negotiation with decommitment for dynamic resource allocation in cloud computing," In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), vol. 1, 2010,pp. 981-988.

[30] X. Nan, H. Yifeng, G. Ling, "Optimal resource allocation for multimedia cloud based on queuing model," IEEE 13th International Workshop on Multimedia SIGNAL Processing (MMSP), October 2011, pp. 1-6.

[31] "OpenStack - Open source software for building private and public clouds," Online: http://www.openstack.org

[32] S.Kuribayashi, "Optimal Joint Multiple Resource Allocation Method for Cloud Computing Environments," International

Journal of Research and Reviews in Computer Science (IJRRCS), Vol.2 No.1, February 2011, pp. 1-8.

[33] M. Al-Fares, A. Loukissas, A. Vahdat, "A scalable, commodity data center network architecture," Proceedings of the ACM SIGCOMM 2008 conference on Data communication, August 2008.

[34] H. Goudarzi, M. Pedram, "Multi-dimensional SLA-Based Resource Allocation for Multi-tier Cloud Computing Systems," IEEE International Conference on Cloud Computing (CLOUD), July 2011, pp. 324-331.

[35] D. Dib, N. Parlavantzas, C. Morin, "SLA-Based Profit Optimization in Cloud Bursting PaaS," 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2014, pp. 141-150.

[36] F. Pires, B. Baran, "Multi-objective Virtual Machine Placement with Service Level Agreement: A Memetic Algorithm Approach," IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC), 2013, pp. 203-210.

[37] O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, Y. Coady, "Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis," IEEE 3rd International Conference on Cloud Computing (CLOUD), July 2010, pp. 91-98.

[38] Y. Awano, S. Kuribayashi, "A joint multiple resource allocation method for cloud computing environments with different QoS to users at multiple locations," IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), 2013, pp. 1-5.

[39] F. Wuhib, R. Stadler, H. Lindgren, "Dynamic resource allocation with management objectives—Implementation for an OpenStack cloud," 8th international conference and 2012 workshop on systems virtualiztion management (svm), Network and service management (CNSM), October 2012, pp. 309-315.

[40] ETSI GS NFV 002 V1.2.1 (2014-12), "Network Functions Virtualisation (NFV); Architectural Framework," Online: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

[41] ETSI GS NFV 003 V1.2.1 (2014-12), "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV," Online: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf

[42] ETSI, "NFV Whitepaper #3," Oct 14, 2014, Online: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[43] ETSI GS NFV-SWA 001 V1.1.1 (2014-12), "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture," Online: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_nfv-swa001v010101p.pdf

[44] Recommendation ITUT Y.3510 (05/13): "Cloud computing infrastructure requirements."

[45] A. Belbekkouche, M. Hasan, A. Karmouch, "Resource discovery and allocation in network virtualization," Communications Surveys & Tutorials, IEEE 14.4 (2012): 1114-1128.

[46] Y. You, M. Zulkerine, A. Haque, "A distributed Defense framework for flooding-based DDOS attack" Third International conference on Availability and security, 2008.

[47] P. Cramton, Y. Shoham, R. Steinberg, editors, "Combinatorial Auctions," MIT Press, 2006.

[48] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, C. Meirosu, "Research Directions in Network Service Chaining," IEEE SDN for Future Networks and Services (SDN4FNS), November 2013, vol., no., pp.1-7.

[49] C. Peltz, "Web Services Orestrestration and Choreography," Computer, vol. 36, no. 10, pp. 46-52, Oct. 2003.

[50] W. Lin et al., "Dynamic Auction Mechanism for Cloud Resource Allocation," 10th International Conference on Cluster, Cloud and Grid Computing, IEEE/ACM, 2010, pp.591-592.

[51] M. Rost and S. Schmid, "Service chain and virtual network embeddings: Approximations using randomized rounding," CoRR, vol. abs/1604.02180, 2016. [Online]. Available: http://arxiv.org/abs/1604.02180

[52] D. Gmach, J. Cherkasova, "Satisfying service level objectives in a self-managing resource pool," In Proc. Third IEEE international conference on self-adaptive and self-organizing system (SASO'09), 2009, pp: 243-253.

[53] Z. Kong et.al, "Mechanism Design for Stochastic Virtual Resource Allocation in Non-Cooperative Cloud Systems," IEEE 4th International Conference on Cloud Computing, 2011, pp: 614-621.

[54] F. Wuhib, R. Stadler, "Distributed monitoring and resource management for Large cloud environments," IFIP/IEEE International Symposium on Integrated Network Management (IM), 2011, pp: 970-975.

[55] L. Wang, Y. Lu, "Efficient power management of heterogeneous soft real-time clusters," In Proceedings of IEEE Real-Time Systems Symposium, Barcelona, Spain, Dec. 2008.

[56] K. H. Kim, A. Beloglazov, R. Buyya, "Power-aware provisioning of Cloud resources for real-time services," In Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science (MGC '09), ACM, New York, NY, USA, 2009, pp: 1-6.

[57] N. Kappiah, V. Freeh, D. Lowenthal, "Just In Time Dynamic Voltage Scaling: Exploiting Inter-Node Slack to Save Energy in MPI Programs," Supercomputing, Proceedings of the ACM/IEEE SC 2005 Conference, Seattle, USA, Nov. 2005, pp: 12-18.

[58] A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," Communications Surveys & Tutorials, IEEE 16.3, 2014, pp. 1617-1634.

[59] D. Roman et al., "Web Service Modeling Ontology," Applied ontology 1.1 (2005): 77-106.

[60] K. Pentikousis, Y. Wang, W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," Communications Magazine, IEEE, July 2013, pp: 44-53.

[61] E. Cronin, S. Jamin, J. Cheng, A. Kurc, D. Raz, Y. Shavitt, "Constrained mirror placement on the Internet," IEEE Journal on Selected Areas in Communications archive, Vol.20 no. 7, September 2006, Page 1369-1382.

[62] D. Bhamare, R. Jain, M. Samaka, G. Vaszkun, A. Erbad, "Multi-Cloud Distribution of Virtual Functions and Dynamic Service Deployment: OpenADN Perspective," 2015 IEEE International Conference on Cloud Engineering.

[63] Recommendation ITUT Y.1564 (03/11): "Ethernet service activation test methodology." Online: https://www.itu.int/rec/T-REC-Y.1564-201103-I/en.

[64] D. Qiang, Y. Yan, A. Vasilakos, “A survey on service-oriented network virtualization toward convergence of networking and cloud computing,” Network and Service Management, IEEE Transactions on 9.4 (2012): 373-392.

[65] A. Hakiri, A. Gokhale, P. Berthou, D. Schmidt, T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," Computer Networks 75, 2014, p.453-471.

[66] N. Chowdhury, R. Boutaba, “A survey of network virtualization,” Computer Networks 54.5, 2010, pp. 862-876.

[67] F. Paraiso, N. Haderer, P. Merle, R. Rouvoy, L. Seinturier, “A Federated Multi-cloud PaaS Infrastructure,” IEEE 5th International Conference on Cloud Computing (CLOUD), 2012, pp.392-399.

[68] J. Simarro, R. Moreno-Vozmediano, R. Montero, I. Llorente, “Dynamic placement of virtual machines for cost optimization in multi-cloud environments,” in High Performance Computing and Simulation (HPCS), 2011 International Conference on , 2011, pp.1-7.

[69] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, “A view of cloud computing,” Communication, ACM, April 2010, pp. 50-58.

[70] H. Li et. al, “Service Function Chaining (SFC) Control Plane Components & Requirements,” Internet-Draft, draft-ietf-sfc-control-plane-02, November 2015, p.27.

[71] S. Lee, S. pack, M. Shin, E. Paik, R. Browne, “Resource Management in Service Chaining,” Internet-Draft, draft-irtf-nfvrg-resource-management-service-chain-02, October 2015, p.17.

[72] N. Figueira et. al, “Policy Architecture and Framework for NFV Infrastructures,” Internet-Draft, draft-irtf-nfvrg-nfv-policy-arch-02, October 2015, p.23.

[73] A. Luigi, A. Iera, G. Morabito, “The internet of things: A survey,” Computer networks, 2010, p.2787-2805.

[74] M. Daniele, et al. “Internet of things: Vision, applications and research challenges,” Ad Hoc Networks 10.7 (2012), p.1497-1516.

[75] A. Ian F., S. Lin, P. Wang. "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: an overview and qualitative evaluation," Computer Networks, 2015, p.66-79.

[76] S. Sahel, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, P. Demeester, “Network service chaining with optimized network function embedding supporting service decompositions,” Computer Networks, 2015, p.492-505.

[77] C. Guozhen, H. Chen, H. Hu, Z. Wang, J. Lan, “Enabling network function combination via service chain instantiation,” Computer Networks, 2015, p.396-407.

[78] P. Wang, J. Lan, X. Zhang, Y. Hu, S. Chen, “Dynamic function composition for network service chain: Model and optimization,” Computer Networks 92, 2015, p.408-418.

[79] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. Turck, R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” Communications Surveys Tutorials, IEEE, Accepted for Publication September 2015.

[80] M. Bouet, J. Leguay, T. Combe, V. Conan, “Cost based placement of vDPI functions in NFV infrastructures,” International Journal of Network Management 25.6, 2015, p.490-506.

[81] Y. Li, M. Chen, “Software-Defined Network Function Virtualization: A Survey,” in Access, IEEE , vol.3, no., 2015, p.2542-2553.

[82] P. Quinn, J. Guichard, “Service Function Chaining: Creating a Service Plane via Network Service Headers,” Computer, vol.47, no. 11, Nov. 2014, p.38-44.

[83] “IoT report by Gartner, inc.,” [Online]: http://www.gartner.com/newsroom/id/2970017

[84] M. Luizelli, L. Bays, L. Buriol, M. Barcellos, L. Gaspary, “Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions,” IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015.

[85] J. Ferrer Riera, X. Hesselbach, E. Escalona, J. Garcia-Espin, E. Grasa, “On the complex scheduling formulation of virtual network functions over optical networks,” 16th International Conference on Transparent Optical Networks (ICTON), 2014, July 2014, pp. 1–5.

[86] S. Mehraoftaghdam, M. Keller, H. Karl, “Specifying and placing chains of virtual network functions,” 3rd International Conference on Cloud Networking (CloudNet), 2014 IEEE, Oct 2014, pp. 7–13.

[87] J. Moura, D. Hutchison, “Review and analysis of networking challenges in cloud computing,” Journal of Network and Computer Applications 60 (2016), pp. 113-129.

[88] W. Li, Y. Zhong, X. Wang, Y. Cao, “Resource virtualization and service selection in cloud logistics,” Journal of Network and Computer Applications, 36(6), pp. 1696-1704.

[89] H. Kim, N. Feamster, “Improving network management with software defined networking,” IEEE Communications Magazine, vol. 51, no. 2, 2013, pp. 114-119.

[90] M. Anwer, M. Motiwala, M. bin Tariq, N. Feamster, "SwitchBlade: a platform for rapid deployment of network protocols on programmable hardware," In Proceedings of ACM SIGCOMM, 2010.

[91] N. Feamster, J. Rexford, E. Zegura, “The road to SDN,” Queue, vol. 11, no. 12, Dec. 2013, pp. 20:20–20:40.

[92] T. Anderson, L. Peterson, S. Shenker, J. Turner, “Overcoming the Internet impasse through virtualization,” Computer 4, 2005, pp. 34-41.

[93] K. Oberle, M. Kessler, M. Stein, T. Voith, D. Lamp, S. Berger, “Network virtualization: The missing piece,” In Intelligence in Next Generation Networks, 2009, ICIN 2009, 13th International Conference on, IEEE, pp. 1-6.

[94] Q. Zhang, L. Cheng, R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” Journal of Internet services and applications 1.1, 2010, pp. 7-18.

[95] A. Bremler-Barr, Y. Harchol, D. Hay, “OpenBox: Enabling Innovation in Middlebox Applications,” Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization, ACM, 2015.

[96] H. Hawilo, A. Shami, M. Mirahmadi, R. Asal, “NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC),” Network, IEEE, 28(6), 2014, pp. 18-26.

[97] J. Park et al., “Locality-aware dynamic VM reconfiguration on MapReduce clouds,” In Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing, ACM, 2012, pp. 27-36.

[98] R. Cohen, L. Lewin-Eytan, J. S. Naor and D. Raz, “Near-optimal placement of virtual network functions,” IEEE Conference on Computer Communications (INFOCOM), Kowloon, 2015, pp. 1346-1354.

[99] M. Bari et al., “On orchestrating virtual network functions,” In Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM) (CNSM '15), IEEE Computer Society, Washington, DC, USA, 2015, pp. 50-56.

[100] H. Moens, F. Turck. "VNF-P: A model for efficient placement of virtualized network functions," Network and Service Management (CNSM), 2014 10th International Conference on. IEEE, 2014.

[101] M. Bagaa, T. Taleb, A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud"' In Wireless Communications and Networking Conference (WCNC), 2014 IEEE, pp. 2402-2407.

[102] M. Bouet, J. Leguay, T. Combe, V. Conan, "Cost - based placement of vDPI functions in NFV infrastructures," International Journal of Network Management, 2015, pp. 490-506.

[103] Z. Qazi, C. Tu, L. Chiang, R. Miao, V. Sekar, M. Yu, "SIMPLE-fying middlebox policy enforcement using SDN," in Proc. of the ACM SIGCOMM'13, Hong Kong, China, August 12-16, 2013, pp. 27-38.

[104] A. Basta et al., "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem," Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges. ACM, 2014, pp. 33-38.

[105] “Open Source Mano (OSM)”, 2016. Onile: https://osm.etsi.org/

[106] “Carrier Ethernet and SDN”, MEF, 2014. Online: https://www.mef.net/Assets/White_Papers/Carrier_Ethernet_and_SDN_Part_1_-_An_Industry_Perspective_08-14-14.pdf

[107] “Accelerating NFV Delivery with OpenStack,” OpenStack, 2016. Online: https://www.openstack.org/assets/telecoms-and-nfv/OpenStack-Foundation-NFV-Report.pdf

[108] S. Fayazbakhsh, L. Chaing, V. Sekar, M. Yu, J. Mogul. “Enforcing network-wide policies in the presence of dynamic middlebox actions using FlowTags,” in Proc. of NSDI, 2014.

[109] H. Farhady, L. HyunYong Lee, N. Akihiro, “Software-defined networking: A survey,” Computer Networks 81, 2015, pp. 79-95.

[110] J. Matias, J. Garay, N. Toledo, J. Unzilla, E. Jacob, “Toward an SDN-enabled NFV architecture,” Communications Magazine, IEEE, 53(4), 2015, pp. 187-193.

[111] M Ghaznavi, N. Shahriar, R. Ahmed, R. Boutab, “Service Function Chaining Simplified,” arXiv:1601.00751, 2016.

[112] L. Battula, “Network Security Function Virtualization(NSFV) towards Cloud computing with NFV Over Openflow infrastructure: Challenges and novel approaches,” Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, New Delhi, 2014, pp. 1622-1628.

[113] S. Scott-Hayward, G. O'Callaghan, S. Sezer. “SDN security: A survey,” Future Networks and Services (SDN4FNS), 2013 IEEE SDN For. IEEE, 2013, pp. 1-7.

[114] T. Zahir, “Security and Privacy in Cloud Computing,” IEEE Cloud Computing 1.1, 2014, pp. 54-57.

[115] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," Communications Magazine, IEEE, 51(7), 2013, pp. 36-43.

[116] E. Rescorla, “SSL and TLS: designing and building secure systems,” Vol. 1. Reading: Addison-Wesley, 2001.

[117] T. Dierks, “The transport layer security (TLS) protocol version 1.2.,” 2008.

[118] S. Vaudenay, “Security Flaws Induced by CBC Padding—Applications to SSL, IPSEC, WTLS...,” Advances in Cryptology—EUROCRYPT, Springer Berlin Heidelberg, 2002, pp. 543-545.

[119] H. Basilier, M. Darula, J. Wilke, “Virtualization network services—the telecom cloud”, Ericsson Review, March 28, 2014, pp. 2-9.

[120] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in 2014 IEEE Network Operations and Management Symposium (NOMS), May 2014, pp. 1-9.

[121] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on, Oct 2015, pp. 255-260.

[122] T. Lukovszki and S. Schmid, "Online admission control and embedding of service chains," in In Proceedings of SIROCCO, 2015.

[123] T. Kuo, B. Liou, J. Lin, and M. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in IEEE International Conference on Computer Communications (INFOCOM 2016), San Francisco, USA, April 2016.