IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 17, 2021, accepted April 5, 2021, date of publication April 8, 2021, date of current version June 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071649

Softwarization of 5G Networks–Implications to Open Platforms and Standardizations

DAVID LAKE[1], NING WANG[1], (Senior Member, IEEE), RAHIM TAFAZOLLI[1], (Senior Member, IEEE), AND LOUIS SAMUEL[2]

[1]5G Innovation Centre, Institute for Communication Systems, University of Surrey, Guildford GU2 7XH, U.K.
[2]Emerging Technologies and Incubation Group, Cisco Systems, London EC2M 7EB, U.K.

Corresponding author: David Lake (d.lake@surrey.ac.uk)

This work was supported in part by the University of Surrey's 5G Innovation Centre 940 (5GIC) (http://www.surrey.ac.uk/5gic), and in part by the EPSRC NG-CDI under Grant EP/R004935/1.

ABSTRACT Softwarization has been deemed as a key feature of 5G networking in the sense that the support of network functions migrates from traditional hardware-based solutions to software based ones. While the main rationale of 5G softwarization is to achieve high degree of flexibility/ programmability as well as reduction of total cost of ownership (TCO), it remains an interesting but significant issue on how to strike a desirable balance between system openness and necessary standardization in the context of 5G. The aim of this article is to systematically survey relevant enabling technologies, platforms and tools for 5G softwarization, together with ongoing standardization activities at relevant SDOs (Standards Developing Organizations). Based on these, we aim to shed light on the future evolution of 5G technologies in terms of softwarization versus standardization requirements and options.

INDEX TERMS 5G, SDN, NFV, softwarization, network slicing, network flexibility.

I. INTRODUCTION

The legacy 4G Long Term Evolution (LTE) Evolved Packet Core (EPC) has typically been built around a fixed architecture using ''black box'' technologies. Whilst such a strategy has been adequate for supporting a relatively limited range of applications in 4G, system rigidity apparently has become a bottleneck feature for emerging 5G-oriented services with a much wider variety of requirements and characteristics. Consequently, the move to softwarization of the network layer in 5G has seen a trend to replace the legacy hardware supported network functions with software based ones. These range from network programming to open versions of the network hardware all with the aim of achieving a higher level system programmability than is possible with fixed, for-purpose solutions. On the other hand, by its nature, standardization seeks to impose a unified structure and regulation in order to assure system interoperability, which is understandably essential for traditional networks realized with hardware black boxes. With softwarized 5G networking, it can be envisaged that there is a tradeoff between softwarization and standardization which impacts the fundamental degree of system flexibility that can be achieved.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaofei Wang.

A. SCOPE

In the context of 5G, this survey paper addresses three key questions in the move to softwarization of the network:

1) SOFTWARIZATION VS. PERFORMANCE

First, the paper discusses the trade-off between the current 5G softwarization in order to achieve the flexibility required and potential deterioration in terms of packet forwarding performance in the user plane. Whilst these effects can be mitigated against in a number of ways such as increased compute resources, under-subscribing of services, splitting of functionality across several software instances, the question remains as to whether the cost and complexity involved in these mitigation methods in the name of increased flexibility offers advantages over current hardware-only solutions. Additionally, ''hardware-assisted'' or hybrid techniques are now in-use which offer somewhat less flexibility than an all-software solution. Unlike other services which consume data in the North-South direction [1], the 5G core may place the User Plane Functions across the East-West plane; recognizing

and quantifying corner-cases where performance may be impacted in such a software-only solution is important.

2) SOFTWARIZATION VS. STANDARDIZATION

Whilst standards are important to ensure interoperability, there is an inevitable trade-off in terms of flexibility; the key promise of softwarization is that of rapid innovation and therefore flexibility whereas a totally standards-based approach would move solutions more towards a harmonized and therefore less flexible set of outcomes. Driving these different approaches are a set of SDOs, each with a different set of goals, scopes and operational procedures. At times these can be seen to be both by-turns complementary and competing in specific overlapping areas. As a second question, the paper therefore considers how the operation of these standards groups acts on the development of the key technology enablers and directs architectures asking how that influences system flexibility both positively and negatively.

Without an over-arching standard, there is a risk of ''point-solutions'' being designed which are difficult to integrate to existing frameworks and therefore ultimately become inflexible. At present, there are a number of such over-arching architecture-lead initiatives which often consume the same input projects but address different use-case scenarios – for example both the OPNFV and CORD architectures use the OpenStack virtualization project

Whilst this is desirable for the specific use-case, the potential exists for the software development efforts in one of the donor projects to become too specifically driven by one particular use-case. It would be useful to see more consolidation between architectural endeavors that are addressing very similar or highly complementary use-cases. This is especially true as the fixed-line and mobile packet core environments themselves become a single, consolidated entity.

3) OPEN-SOURCE VS. ''CLOSED''-SOURCE SOFTWARIZATION

Finally, the paper asks how the different models of softwarization are encountered in these projects including the role of open-source software remembering that open-source does not necessarily imply freely available but more talks about the availability of the source-code for further development. As networking has evolved from simple command-line device-level programmability of the configuration to today's fully programmable techniques including the ability to support custom-designed protocols, projects have taken differing views on the level of openness, ranging from the totally closed, black-box systems that expose an API for programmability to fully open-source and levels in-between. This paper discusses the concept of ''openness'' in the software projects studied and discusses how this impacts on the key concern of flexibility in 5G Networks.

In considering these three different aspects, an example solution is presented in Section 3, sub-section E, which, whilst conforming to prevailing standards, shows how both flexibility and high forwarding performance can be obtained within the bounds of the standards by using components from both the fully open-source and partially open-source communities.

Section 2 discusses the enablers both from an architectural and a technology point-of-view, both of which underpin the 5G core. Section 3 details the existing tools available to build these architectures together with a survey of the Standards Organisations including the method by which they work and their relationship to each other. The survey paper concludes with Section 4 with a summary on the key observations and open issues.

B. MOTIVATION

Prior surveys cover a number of aspects of virtualization [2], [7], [11], network softwarization [3]–[11], Virtual Network Function (VNF) placement [3], [5], [6], [8]–[11], network architecture [7], [11] and [11] touches on some of the standardization efforts, but none detail the performance and flexibility trade-offs or detail the relationship between the standards bodies and which groups are active not only in 5G standardization but also in related protocol work.

This paper draws together this wide topic by considering in a systems-level approach how a combination of different techniques can be used to deliver a flexible, performant, softwarized 5G solution and helps the reader choose the appropriate technology.

II. REPRESENTATIVE 5G SOFTWARIZATION FEATURES AND ENABLERS

A. HIGH LEVEL VIEW OF 5G ARCHITECTURE STRATEGY

Two elements make up the core architecture of the 5G Core – the physical separation of mobile control plane and data plane and the decomposition of service functions as defined by the 3rd Generation Partnership Project (3GPP) in TS 23.501 [12]. Whilst the essential concepts of the 5G Core have been defined, the technology and tools are left up to implementation as 3GPP definitions are at the interface level rather than at the individual component level. Completion of the work at 3GPP therefore does not mean that all network functions are operational, but simply that the target architecture is set.

By separating out the control and data-plane network functions, placement of the relevant service to provide the required service level can be achieved in a manner which is not possible in the legacy 4G networks. For example, an IoT service may require a very low-latency session – moving the User Plane Function (UPF) and the application connected with this service close to the Radio Access Network (RAN) attachment point will allow the infrastructure to adapt to this requirement. On the other hand, generic Internet access for email does not have such a stringent latency requirement and in this case, the UPF can be centrally located. In both cases, other control-plane elements can be similarly positioned as needed. This softwarization of the packet core is supported by a number of network features as discussed in Section II.

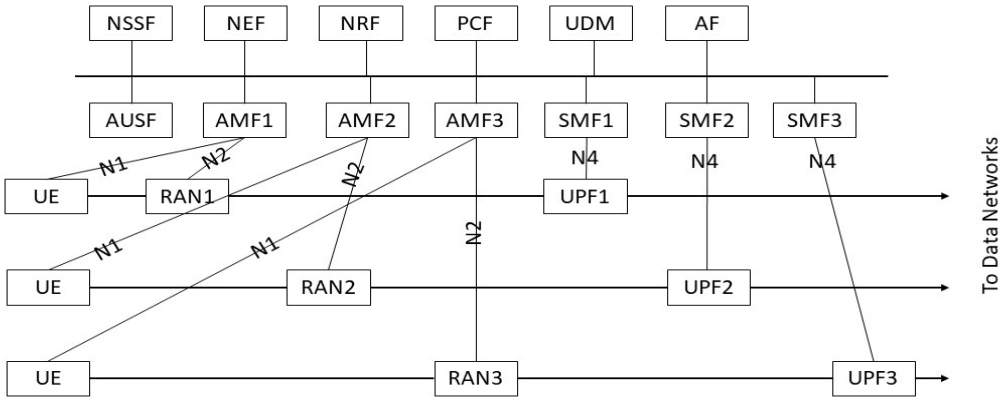

FIGURE 1. Sliced 5G network.

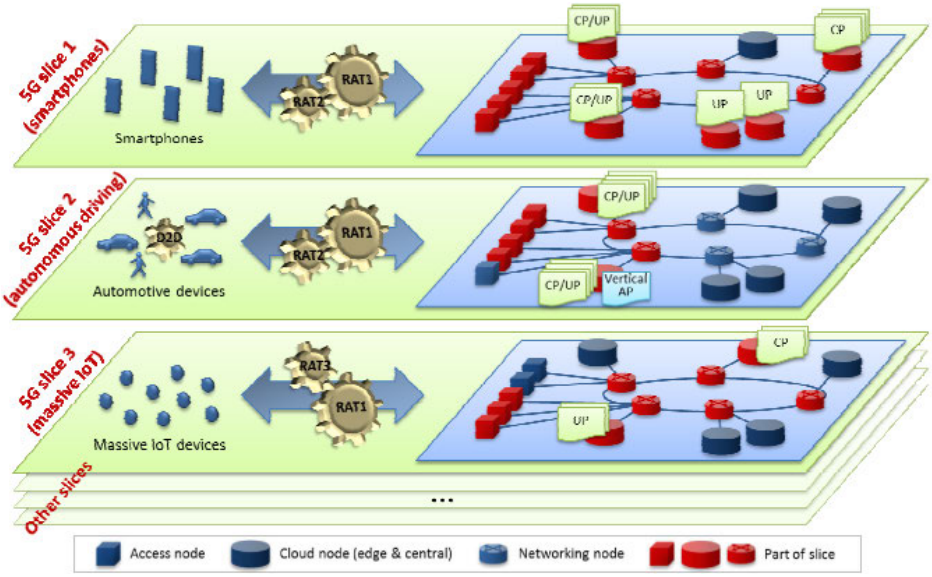

FIGURE 2. Network slicing example [15].

B themselves made possible through technology enablers as discussed in Section II.C.

B. 5G SOFTWARIZATION FEATURES

1) NETWORK SLICING

The term Network Slicing as used in the mobile environment is synonymous with the concepts of Cloud Computing where the one set of dedicated resources provides tailed services for specific applications. In the case of Cloud Computing, this is typically a common network layer with different, centrally located application services although as Foukas *et al.* [6] state, these concepts need to be developed in order to support both different application and network requirements including a mixture of centralized and localized computing capabilities. Figure 1 shows how the static components of the 5G network core may be mapped to multiple network slices.

Conteras and Lopez [13] discuss how the concept of slicing enables operators to provide the physically combined but logically separate sets of infrastructure to meet the competing needs from the user applications. Complete instances of the 5G network can be built within a single slice sharing a single set of infrastructure. The authors further consider how control over the slices could lay outside the traditional service provider. However, challenges remain with defining and controlling the network slices. In order for the user to be unaware of the other slices sharing the infrastructure, sufficient isolation between the slices must exist as must assurance that the required service can be delivered. Davies and Thompson [14] consider how the ability to trade requirements and resources will be essential and how that impacts on the monitoring of network and compute assets in order to create an holistic view of performance.

NEC's white-paper [15] shows a practical application of Network Slicing in the 5G Core, Figure 2, where a single set of infrastructure appears as independent networks and applications to the use-cases. The ''Next Generation Mobile Networks (NGMN) View on 5G Architecture'' [16]

pictorially explain the concept of slicing well. In reviewing the current state of the 5G Architecture – Gupta and Jha [17] note that whilst the overall architecture is understood and the interface points defined in 3GPP TS23.501 [12], many aspects of the operation and optimization of the core network remain unresolved.

Beyond 3GPP Release 16, scope exists to identify use-cases not currently supported by the current 5G architecture. If the time is taken now to build a flexible core solution, deploying as-yet undefined use-cases and protocols will be possible given the software-based nature of the components.

Looking beyond the mobile environment, given the inherent cross-domain nature of the 5G Core proposals, future service provider networks could be comprised of wireless and wireline access on the same infrastructure. Projecting forward, there could be a single set of physical infrastructure (geographically, nationally) with each operator existing simply as a slice or set of slices. This raises some interesting questions about ownership and revenue settlement in future service-delivery; for example, could user service contracts be with the content owner who in-turn settles transport charges across any application and network path as needed rather than the user having to pay a separate access and content charge. This would be a similar model to the electricity supply industry in some countries (e.g. UK, Canada) where power is bought from third parties who settle payment through the various transport networks involved in supply.

2) SFC

Service Function Chaining (SFC) is a technology which allows both topological and transport independence from the underlying hardware by defining the end-to-end delivery of services through an ordered set of defined service functions [18]. Farrel [19] discusses how the goals of Network Slicing and SFC are analogous and therefore complementary in delivering defined outcomes. Filsfils *et al.* [20], show how SFC is seen as a key enabler for the delivery of network services within a substrate built of virtualized functions. Halpern and Pignataro [18] detail how the operation of SFC depends on the ability to classify traffic along the service path using a logically separated control and data plane but without specifying exactly how this logical separation may be realized. At a 3GPP Plenary in 2017, Cooper [21] presented areas in Internet Engineering Task Force (IETF) of relevance to 5G networks – Service Chaining was described as a core networking feature able to support the 5G sliced environment.

Classification and reclassification of traffic can occur at any point in the network – Quinn and Guichard [22] discuss the challenges of choosing a suitable method of carrying classification between devices and between different vendor's hardware showing how the use of a Network Service Header (NSH) can be added to any packet as an inter-SFC-enabled marker. Quinn *et al.* [22] explain how such headers can be inserted from either a centralized or distributed control plane using the Software-Defined Networking (SDN) paradigm including extensions to SDN controllers such as OpenDaylight and software switches such as Open vSwitch. Haeffner *et al.* [23] explain how NSH could be determined from the Policy and Charging Rules Function (PCRF) in 4G networks but the example concentrates more on defining a service path rather than native end-point mobility.

An issue with NSH is the inability to use the underlying IP routing techniques of the network such as Equal Cost Multi-Path – a solution to this is proposed by Filsfils *et al.* [20] in the use of the source-routing based Segment Routing (SR) scheme which uses a push/pop technique to add/remove Segment IDs with local significance. A benefit to this approach over NSH is that the SR technique is a retro-fit to an existing IP-based network complete with in-place complex routing and traffic engineering schemes, although it would require modification of the elements to be able to handle the insertion and removal of the SIDs. SR can also operate in a controller-based SDN method – Filsfils *et al.* [20] detail how a network topology can be built and disseminated with OpenDaylight using the Path Computation Element Protocol (PCEP). NSH also provides metadata to the control plane to indicate suitability/operation of a network path.

All SFC techniques require effective traffic classification. Medhat *et al.* [24] describe how the two competing paradigms, SFC and OpenFlow differ in that the OpenFlow model relies on tuple match/action. OpenFlow switches have subsequently been modified such that they support NSH insertion/deletion although as discussed later, as an OpenFlow switch is stateless, the interaction of a stateful program in the controller would be required.

It is unclear at present how typical operator Service Level Agreements which are held at the Operations Support System/Business Support System (OSS/BSS) layer would be related to network delivery. This requires development of techniques to enable expected outcomes to be mapped to service forwarding decisions. Current management and orchestration system in use in NFV are able to understand orchestration of network service both within an administrative domain and across administrative domains at a systems level, described by Medhat *et al.* [24] as lifecycle management of VNFs, topological view of the data-plane between VNFs – the VNF Forwarding Graphs (VNF-FG) – but instantiation of a specific policy related to delivers of a single user-service is not currently possible.

Related to this is an inability to determine the potential impact on service delivery due to placement of the VNFs - Medhat *et al.* [24] detail how the presence of a bottleneck in the VNFG is currently a catastrophic and indeterminate fault to the orchestration and intent layers of the service. These issues with SFC become more problematic when one considers the highly dynamic state of the mobile packet core especially in relation to the ability for 5G to deliver on the promise of differentiated services against defined outcomes. Farrel [19] limits the applicability of SFC to the backhaul and metro networks in the overall 5G topology.

Discussing the overall impact on the control plane of an SDN-based, edge-focused network, Callegati *et al.* [25] detail how with careful analysis of the network topology, a combination of both legacy forwarding and specific SDN programming can be used to minimize control-plane traffic. However, the study concentrates on a very small proof-of-concept based on a single, centralized controller with Mininet and therefore is not sufficiently representative of large, geographically dispersed mobile networks.

Bhamare *et al.* [26] whilst seeing how SFC can become an all-encompassing service descriptor as well as a transport tool, discuss how SFC is currently not flexible enough to be able to react to changing network attachments – for example as a user moves through the mobile network. Such dynamism would require that the radio and data-network control planes were closely coupled to the point where the SLA was known between the two. Whilst this is the case today for Voice over LTE (VoLTE) where the elements which provide the VoLTE service are closely couple to the RAN via the PCRF, this only exists within the one operator domain. In order to extend to a multi-domain world, mechanisms need to exist to enable network and last-mile capabilities to be related. Bhamare *et al.* goes on to discuss how there is a need for an overall services descriptor language able to express an entire end-to-end service in a manner which allows all elements in the transport to create and maintain the Service Level Agreement (SLA). The authors compare this to the state-of-the-art in web service construction through the use of the Web Service Definition Language (WSDL).

Given that NSH carries metadata related to packet transport already used by the forwarding plane control path, it would seem that extensions to this are required to ensure that all elements such as RAN and server metadata can be accounted for.

C. TECHNOLOGY ENABLERS

1) SDN

The core concept behind SDN is the separation of control and data planes in the data networking environment in order to provide better manageability and easier end-to-end policy management in Enterprise networks. Initial experiments resulted in the Ethane architecture [27] where relatively dumb forwarding devices were placed under control of an intelligent controller which had an over-arching view of the packet path. This separation of control-plane and data-plane and concentration on resource availability through policy was very similar to the International Telecommunications Union (ITU) SS7 system [28] used to carry digital voice in the telephone network. Work on OpenFlow [29] grew from the Ethane work addressing several of these concerns and building a more generalized version of the architecture. Since the first discussion of SDN concerns have been raised regarding the ability of the controller to scale both locally and globally. First, there is a trade-off to be made in terms of the match/action delay on the first frame especially where the controller is at some distance from the switch. A solution would be to pre-populate all policy to the switch but this defeats the object of having a central controller as maintaining network-wide concurrency, a major issue with traditional routing protocols such as Border Gateway Protocol (BGP), is exactly what SDN is designed to resolve. Next is the need to ''age-out'' flows in a timely manner, very relevant in the mobile use case where a user may be moving between attachment points.

Given these issues the open question of the controller architecture in terms of scalability remain. Sezer *et al.* [30] and Sayadi *et al.* [31] both address this question in terms of considering a distributed or hierarchical controller solution. There are trade-offs in terms of latency for packet match/action and system concurrency in every case. Sayadi *et al.* [31] present findings from 5G-NORMA where a hierarchical controller was developed with each SDN controller managing a single network slice.

The Controller Placement Problem remains one of the key unresolved issues in SDN – Wang *et al.* [32] detail 11 on-going research efforts either attempting to find optimal placement based on network topology or to solve other issues associated with various controller layouts such as redundancy, energy efficiency but this topic remains top-of-mind for all current SDN deployments. With particular respect of mobile networks using SDN in the packet core, Selvi *et al.* [33] compare various placement options and are able to quantify some common mobile KPIs in relation to controller placement but do not draw definitive conclusions. The authors state that controller placement remains a key unanswered item in the design of Software Define Mobile Networks.

An interesting emerging approach to this issue is the use of Artificial Intelligence to enhance the operation of many aspects of the SDN infrastructure including the controller placement problem. Latah and Toker [34] survey the current active research in this area, citing a group currently active in the use of data-driven machine learning techniques to predict optimal controller placement for future traffic [35]. Given that much data is available in the mobile network already regarding user placement both presently and historically, this data-set could make an interesting input to help predict future topology. As P4 [36] programmable hardware switches have become available including the ability to dynamically program the datapath using P4Runtime, SDN has expanded to include P4Runtime as a southbound API in addition to OpenFlow. P4 and P4Runtime are discussed in greater detail in Section III.B.

As a key component in the goal of delivering flexibility in softwarized networks, it is important to consider the journey from the past, for-purpose networking hardware to the network-wide programmable solutions available today. Table 1 compares the evolution of networking programmability.

2) NFV

A key enabling technology for future mobile network is Network Function Virtualization. Dating back to a white paper

TABLE 1. Evolution of network programmability.

	Traditional Networking	OpenFlow	P4
Configuration	Stored on device (CLI)	Split between fixed config on device and controller programming	Defined on controller and sent to device
Protocol Support	Determined by hardware	Fixed pipelines	Defined by P4 programming (C-like representation of protocol)
Dynamic Programmability	Typically not possible	Packet Match/Action rules can divert to controller for new flows	P4Runtime can dynamically program hardware packet treatment
End-user Extensibility	Not possible	Possible, but pipelines controlled by OpenFlow release control	P4 allows any packet treatment to be written in hardware
SDN Capabilities	Minimal – NetConf/YANG representation of devices	SDN controller required to handle all unknown flows	P4 program can be stand-alone with SDN controller augmenting treatment via P4Runtime
Speed	Purpose-build hardware – can be fast	Fast for hardware flows; slower if packets diverted to controller	Fast – P4 programs the hardware directly

presented at the ''SDN and OpenFlow World Congress'' in Darmstadt, Germany in October 2012 [37], the basic premise is that by using a combination of compute virtualization - such as has become prevalent in Enterprise and Cloud computing environments – and SDN techniques such as OpenFlow, traditional for-purpose networking equipment in the core of the Service Provider network can be replaced by relocating the network processing element to software running within Virtual Machines. As stated by Briscoe *et al.* [37], SDN is an integral part of NFV and NFV can be seen as an extension to the SDN landscape.

A proof-of-concept was constructed moving a Digital Subscriber Line (DSL) Broadband Remote Access Server (B-RAS) which terminates Point-to-Point Protocol (PPP) over Ethernet frames from home DSL services into a combination of a software control plane running in a Virtual Machine and OpenFlow switches modified to terminate the PPP tunnel. Whilst the system was tested for functionality and rough performance determined, none of the other claimed aspects of NFV could be quantified from this technology demonstrator.

It should further be noted that from the Proof-of-Concept built in March 2012 by Briscoe *et al.* [37] and presented to the Internet Research Task Force (IRTF) Software Defined Networking Research Group (SDNRG) in 2013, it is not clear as to whether a virtual compute or bare-metal system was used to host the PPP function. The virtualization here was a decomposition of network elements than a move to a virtual machine environment. There may have been some ''over-reading'' of this lab work and mis-association of terminology by readers. Not all of the benefits of NFV have been realized in the 6 years since the initial presentation and issues around achieving the required performance levels have dominated research work as discussed later in this chapter.

Another aspect of Enterprise virtualization which was attractive to the team was the ability to operate multi-tenancy systems – one set of physical infrastructure shared between two-or-more tenants each of whom see only the resources allocated to them and are otherwise unaware that they are using shared infrastructure (Figure 3). In the Service Provider core, Chiosi *et al.* [38] envisage that such multi-tenancy would be network slices, logically separated end-to-end networks in some respects similar to Virtual Routing and Forwarding (VRF) in MPLS but with the addition of application functions. This would enable one set of physical infrastructure to be split into multiple slices, each providing a specific set of service levels for the users. Chiosi *et al.* [38] discuss how such softwarization will lead to ''flexibility'' but make no connection to the potential pit-falls of such virtualization such as overall system performance.

Several virtualization methods exist to enable such multi-tenancy and is discussed in detail by Kominos *et al.* [39]. To-date, other than a number of proprietary systems such as Xen and VMWare, the main open-source virtualization system is OpenStack. Through its linkage to the main SDN controllers such as Open Daylight and Open Networking Operating System (ONOS), a combined OpenStack/SDN solution provides the infrastructure in many NFV solutions.

More recently, in order to address the complexity and performance issues associated with OpenStack [39], [40] container technology has come to the fore finding a niche in areas

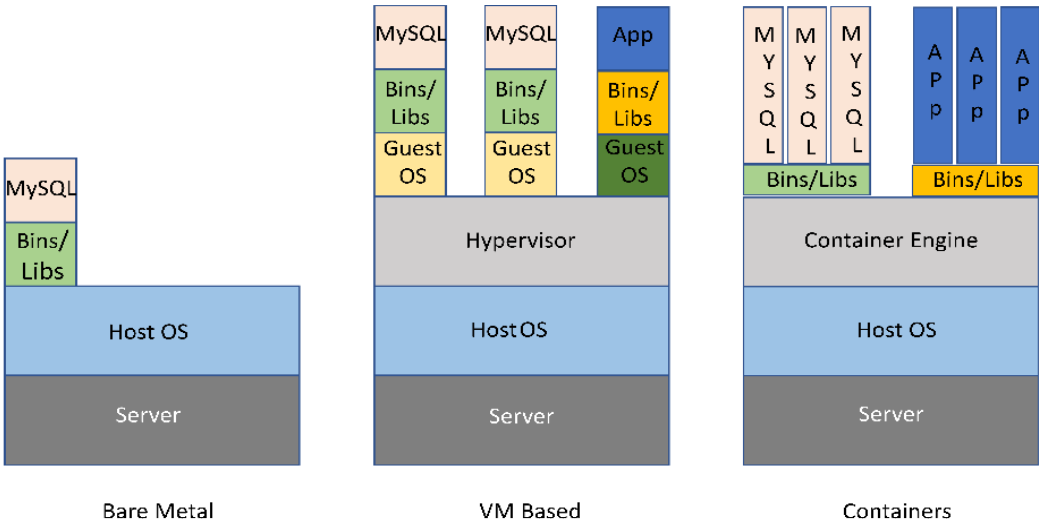

FIGURE 3. Comparison of virtualization options.

where highly efficient use of compute resources with more flexibility than is available from directly-installed bare-metal systems is required, for example at the edge of the access network. Both Kominos *et al.* [39] and Kakakhel *et al.* [40] find that container performance approaches that of bare-metal but with added complexity for Virtual Network Functions (VNFs) which need to interact with the network layer as discussed later in this section. OpenStack can be configured to support container instances instead of the more usual KVM-based full hypervisor. Within the mobile packet core for 5G and beyond, the multi-tenancy solution is seen as analogous to the Network Slicing portions of the 5G system architecture with the same goal of allowing a single set of hardware to appear as several different, logically-separated environments. The MANO framework originally developed as part of the European Telecommunications Standards Institute (ETSI) NFV work which grew from the Clarke presentation is applicable for instantiation and operation of elements in a 5G sliced network core as in essence, the 5G core is a use-case of the overall NFV architecture. Experiences of building 5G sliced networks in this manner are discussed by Mayoral *et al.* [41] and Lake *et al.* [42].

III. EXISTING SOFTWARIZATION TOOLS AND INITIATIVES

Underpinning developments in future packet core are a number of initiatives covering areas of software and hardware implementations in the network control and user planes. A number of forum and alliances have been formed to manage and direct these efforts – at times, the relationship between each of these groups and the technology which they control is not clear and this chapter aims to clarify the currently-operational groups, their scope and relationship with other parts of the environment.

This section is broken down into descriptions of five main sub-areas; the Control Plane, Data Plane, Dynamic Programmability, Virtualization systems and Architectural elements. It should also be remembered that there is a difference between "Open Source" and "Freely Available" – each does not imply the other. This differentiation is discussed by Peterson [43] and the presented taxonomy details where a solution is available without charge.

The first three sections concentrate on solutions available to create a Software Defined Network. The next section, Virtualization Elements, discusses the options for providing the underlying substrate on which these elements are instantiated. Platforms and Alliances details the initiatives which direct efforts to construct a total solution – for example, OPNFV [44] combines OpenStack virtualization with ONOS SDN to create a reference architecture.

A. CONTROL PLANE

There are a large number of SDN controllers available supporting a wide-range of Northbound and Southbound interfaces, all based on the concept of control and data plane separation. Zhu *et al.* [45] detail the architecture and study the performance of 34 controllers. This paper concentrates on the three most common as found in current SDN solutions.

1) OPENDAYLIGHT

OpenDayLight (ODL) [46] was one of the earliest SDN controllers. Written entirely in Java, it is built in three layers; northbound interfaces such as REST and NETCONF to allow external policy to be implemented, a central core with a Service Abstraction Layer based on OSGi (Open Services Gateway initiative) techniques and a set of southbound plug-ins to interface to programmable network devices.

Each ODL cluster retains a model-based, in-memory view of the network for which is it responsible. ODL natively uses BGP to enable isolated SDN islands to exchange routing topology. Systems may be clustered to enable resilience and

redundancy although the effect of latency and bandwidth delay may limit the geographical reach of the complete solution. ODL is based on the Akka distributed datastore [47]. ODL supports a wide-range of southbound interfaces such as OpenFlow, P4, NETCONF, SNMP, BGP, RESTCONF and PCEP through plugin modules. On the Northbound side, as well as RESTful APIs, the DayLight User Experience (DLUX) presents a graphical interface to visualize network topology.

Because of the ease of integration, ODL has been used extensively as the SDN controller in an NFV stack – native integrations into OpenStack, Kubernetes, OPNFV and ONAP exist to enable a single NFV management stack to control both the virtualization and the network topology.

2) ONOS

The Open Network Operating System (ONOS) [48] is an SDN controller designed to provide the control-plane function of a network formed of multiple forwarding elements such as switches. Architecturally, ONOS is built around a scalable core such that it can be clustered by using an Atomix distributed datastore. This provides some level of geographical-based distribution capabilities to the solution including a concept of redundancy in the event of the failure of a controller in the cluster through the election of a new Master Node. However, the performance of the links between the cluster members is material and will impact on the overall operation of the system and thus geo-redundancy should only be considered across small regions.

ONOS supports a wide range of Southbound and Northbound interfaces. The Southbound interface is pluggable and extensible – APIs including OpenFlow, P4, NETCONF, TL1, SNMP, BGP, RESTCONF and PCEP are currently supported. Northbound interfaces include gRPC and a set of RESTful APIs. The software is supported through the Open Source efforts of the Linux Foundation [49] and a very large community of developers and users.

3) RYU

Written in Python, Ryu [50] is the successor to the first SDN controller, NOX which was written in C++ and its iteration POX, written in Python.

Stancu *et al.* [51] describe how NOX and POX both are limited to OpenFlow 1.0 but POX enables more rapid prototyping. Whilst acknowledging that the wide range of protocols supported by Ryu – including quite rarely, OpenFlow 1.5 – makes it ideal for research and prototyping, they show how it is limited in performance due to the use of Python as opposed to the speed inherent in a C++ or Java implementation. However, the ease of development makes it a popular choice in the research community.

There is no clustering facility natively in Ryu and therefore scalability is limited by the capability of the platform in which it is run. Ryu is built using the Zookeeper [52] distributed coordination solution but does not implement any controller synchronization at present and such solutions would need to be provided by the user.

On the southbound side, Ryu supports OpenFlow, NETCONF, OF-Config and some support of P4. The northbound interface is limited to RESTful APIs only and thus Ryu is less useful in an SDN/NFV stack when compared with ONOS and ODL.

B. DATA PLANE

1) OPENVSWITCH

One of the most important pieces of software in the NFV/SDN stack, OpenVSwitch [53] OVS, is a software-based, programmable network switch implemented with the Linux kernel.

The switch supports a number of critical networking features such as 802.1Q VLAN tagging, QoS extensions and native encapsulation and decapsulation of tunneling protocols such as GRE, VXLAN, Geneve and several others. When run as a kernel-mode program, high performance can be achieved although other techniques such as DPDK and SR-IOV (Single Root, Input/Output Virtualization, Intel's proprietary method of associating PCI assets with software stacks) [54] are required to achieve acceptable throughput in many use-cases. OVS can also be run in user-space to enable the software to be run on systems that do not have a kernel-based implementation. The drawback is one of lesser performance and potential interaction with other userspace-based operations. OVS is programmable from the command-line and from an SDN controller using OpenFlow.

2) BOFUSS/CPQD/OFSOFTSWITCH13

Whilst the kernel-space operation of OVS makes it ideal for production environments the ability to implement new features is complex due to the deep level of kernel programming knowledge required.

Initially started by a collaboration between Ericsson and the Centro de Pequisa e Desenvolvimento em Telecomunicações (CPqD) in Campinas, Brazil, the Basic OpenFlow Userspace Software Switch (BOFUSS) makes as its goal easy extensibility. Fernandes *et al.* [55] describe the gestation and motivation of the project detailing the performance difference between OVS and BOFUSS but also showing how BOFUSS was able to implement OpenFlow 1.2 and 1.3 before OVS due to the modular nature of the solution

They go on to describe 40 research-lead projects which have leveraged BOFUSS as the Data Plane. Of particular interest to the 5G Packet Core use case is the Openstate [56] project which aims to address both the issue of holding distributed state in the network and to enable an extensible hardware-based platform capable of supporting multiple protocols. The initial proof-of-concept described by the authors includes a version of the open-source SDN controller Ryu where the standard OpenFlow protocol has been extended with custom verbs to enable population of an additional table held in hardware in the switch which allows state to be held

against identified flows. This allows applications such as a stateful firewall to be built without having to reference the controller. The authors also state their wish to implement the solution on a hardware-based platform in order to improve performance, but to-date have only built software-based proof-of-concepts.

There are a number of issues with the current OpenState implementation which limits its immediate applicability to mobile networks. Firstly, it currently only acts on the available OpenFlow matching paradigm which does not extend to the TEID or the inner-tunnel details as with the extended Open vSwitch above and as would be required in current 5G UPF implementations.

Secondly, the OpenFlow verbs added by the team are unique to the proof-of-concept and would therefore only be applicable to this particular combination of software and hardware.

Thirdly, whilst the team have extended Ryu to issue state-related information, the only manner by which state may be handled is in the programming of the controller. To be fully useful as a source of network-wide state, the controller would need to be able to understand context set externally and match this with the network-level OpenFlow programming. This ''intent-based'' networking construct is discussed at length by Bifulco and Retvari [57].

Finally, whilst OpenState is applicable to both software and hardware forwarding systems, to-date the team has not found it possible to fully build the solution on hardware.

However, the principle of being able to hold state in the network by using extensions to BOFUSS has direct relevance to the 5G UPF use case and has greatly influenced the prototyping carried out by the authors.

3) NETFPGA

The move to capable, open-source hardware has been spear-headed by the Universities of Stanford and Cambridge's NetFPGA board and the most recent iteration, the NetFPGA-SUME based on the Xilinx Vertex-7 FPGA which Zilberman *et al.* [58] note is capable of supporting up to 100 Gbit/s per port. Developments in higher-level programming languages to enable abstraction of the hardware into a language more commonly programmed have resulted in the emergence of P4 as the data-plane programming language of-choice [59].

Development continues towards a simple transition between P4 and the underlying hardware for a number of different FPGA. Solutions such as P4FPGA [60] and P4-NetFPGA [61] discuss solutions to developing in P4 and having the solutions run on hardware. Benchmarking by Wang *et al.* [60] has shown that solutions developed in P4 and then transitioned to FPGAs perform identically to the same solutions developed natively in a lower-level language.

4) DPDK

Initially a proprietary solution from Intel but now managed under the Linux Foundation, the Data Plane Development Kit (DPDK) [62] is a solution for enhancing the performance of network throughput by bypassing the Linux kernel and effectively connecting the Network Interface Card (NIC) directly to the application in userspace.

There are limitations with this approach in that only a limited number of NICS, initially a few Intel devices, are supported. DPDK only allows a limited number of applications to be directly attached and requires command-line configuration to map the PCI address of the NIC to the application. As such, the total bandwidth of the card is shared between the applications. As discussed earlier, DPDK does not address performance between adjacent applications in the same platform

When connecting to virtualization solutions, DPDK can interact with OVS to provide enhanced performance from NIC to Virtual Machine – this is an instance where the data-forwarding component of OVS is run in userspace. DPDK requires that a portion of memory, Hugepages, be allocated for processing packets outside of the PCI bus. This must be manually configured on the host Linux boot line.

To-date, DPDK has proven to be the most effective solution for accelerating OVS in an SDN/NFV environment.

5) P4

First described by Bosshart *et al.* [63] P4 presents a method of describing a data path in terms of a pipeline. Unlike OpenFlow which uses fixed elements within a pre-defined pipeline, P4 enables the user to define the construct of the pipeline itself. Therefore, it is possible to define protocols headers, lookup tables, actions that are to be carried out, counters, etc. Although the syntax and structure of P4 is in many ways similar to C because the language is specific to packet forwarding planes, it is domain-specific and thus has its own compiler and associated toolchain.

P4 may be used on both software and hardware switches including those that have fixed functions which can be defined as blocks within the P4 language. Thus the user can use P4 to define a ''contract'' between the control-plane and data-plane which includes open elements in the forwarding substrate and proprietary elements provided by the OEM. The first version of P4, $P4_{14}$ in many ways was built on the programmable features of OpenFlow and thus only exposed a limited portion of features from hardware switches to be described in the pipeline.

A major change occurred in 2017 with the publication of the $P4_{16}$ Language Specification [36]. Unlike $P4_{14}$, $P4_{16}$ makes no assumption about the device capabilities – instead, the OEM provides a description of the elements available and allows the programmer to manipulate those through the pipeline programming. This has lead to the development of the Protocol-Independent Switch Architecture (PISA) where a number of OEM-provided elements are programmable within a defined pipeline. The capabilities of the underlying switch is defined by the manufacturer through a provided ''architecture.p4'' file. The dataplane actions programmed to the switch is thus a combination of the user-provided

P4 program and the OEM's architecture description. Whilst $P4_{16}$ represents a major step-forward in the evolution of a truly programmable datapath, the data-forwarding actions are fixed in the programming and changing the operation requires that the switch be reprogrammed.

P4Runtime is the method by which the control and data-planes are disaggregated.

C. DYNAMIC PROGRAMMABILITY

1) OPENFLOW

OpenFlow [64] is the programming protocol between the SDN controller and the switch which enables the promise of Open Networking. Because a range of both software switches such as OVS and commercially-available hardware switches support OpenFlow, it has been the introduction of this protocol and the concept of splitting control and data places that has opened-up the world of Open Networking. Prior to the introduction of OpenFlow, the data-plane would be closely-coupled to the control-plane of the forwarding device with the result that innovation in terms of new features was only possible by the Original Equipment Manufacturer (OEM). With the possibility to mix different data-planes and programmability of the controller, developing new network applications becomes easier to achieve. However, the current versions of OpenFlow only allows manipulation of a subset of network elements, typically those associated with the basic 5-tuple IP packet and a few limited tunneling protocols (not including GTP). Wider support requires new developments on the OpenFlow protocol.

Additionally, the programming paradigms used for OpenFlow are somewhat complex and consequently future development directions for OpenFlow appear to be limited.

2) P4RUNTIME

First defined in January 2019 the P4.org API Working Group [65] created the P4Runtime API to provide the ability for external access to the previously closed control-plane.

Unlike the OpenFlow protocol which is used to program match/action within the IP datapath, P4Runtime defines not only the match/actions but also the function of the pipeline itself. P4Runtime programs the pipeline capabilities in terms of the registers and parsers available in the hardware. This means that with the combination of P4 on a hardware switch, software switch and NIC one is able to create a truly protocol independent, programmable forwarding plane.

P4Runtime uses gRPC [66] as the protocol between the P4 platform and the controller. Additionally, P4Runtime has been integrated as a plugin to several of the previously discussed SDN controllers such as ONOS allowing a completely protocol-independent, programmable network forwarding plane to be constructed whilst providing linkage to the control-planes of the Virtualization stacks. In many respects, OpenFlow can be seen as an IP-limited subset of the combination of $P4_{16}$ and P4Runtime.

As key players in the development of Openflow have moved to academic and industrial posts associated with the future development of P4, it is expected that all future innovation in the programmable dataplane will be concentrated on P4 and not OpenFlow

D. VIRTUALIZATION ELEMENTS

Virtualization of operating systems is a key technology in the development of future network infrastructure and two main solutions exist; OpenStack and Kubernetes.

1) OPENSTACK

OpenStack [67] started as a collaborative effort between Rackspace and Anso Labs to provide an Open-Source alternative to existing commercial cloud services such as Amazon's AWS [68]. The NOVA [69] solution written completely in Python from Anso forms the basis of the Compute function in current versions of OpenStack. The current build of OpenStack consists of 12 individual modules which work together to provide a complete solution. Their relationships and interdependencies are shown in Figure 4.

a) NOVA – the Compute platform which provides hypervisor services

b) Neutron – this services forms the basis of the inter-VM networking services within OpenStack and interfaces to external SDN controllers through a number of plugins.

c) Glance – the image management system

d) Sahara – previously known as Savannah, this module provides integration to data-management solutions such as apache hadoop

e) Ceilometer – responsible for collecting and distributing telemetry from the entire openStack solution

f) Keystone – management of user and instance identity, authentication and authorization including interfaces to external databases via LDAP

g)Heat – the orchestration of all elements in order to deliver the required virtual machine

h) Horizon – the graphical front-end to the management of the infrastructure

i) Swift – a shared data storage resource optimized for fast read/write and a simple API

j) Trove – a shared database service

f) Cinder – a shared block-storage service designed as the system resources (e.g. disk) storage point for virtual machines

l) Ironic – a service to enable bare-metal machines to be built as-if a virtual machine within an openStack environment.

Each of these standalone modules may be placed on any one of a number of systems to build a complete OpenStack cluster, the exact layout of compute and networking services being variable with the minimum requirement being one Controller node and one Compute Node. OpenStack presents the Virtual Machines as completely isolated, standalone VMs, interfacing the rest of the infrastrucutre via emulated NICs and the Neutron Networking service. CPU, memory and network resources within the host operating system are isolated

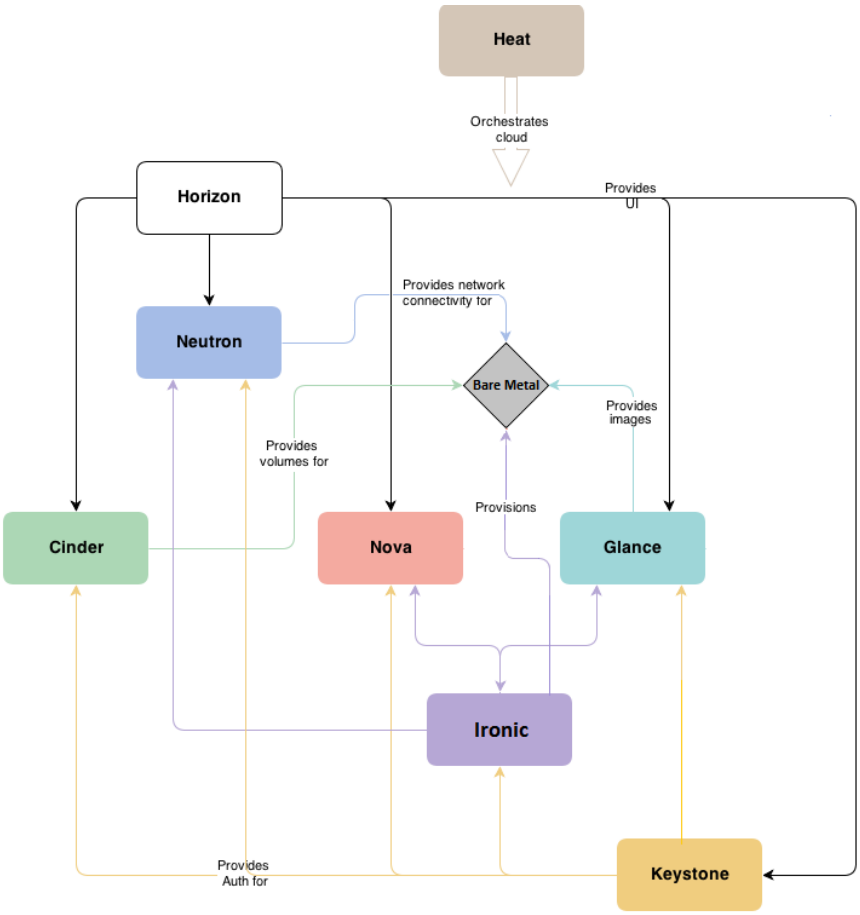

FIGURE 4. Openstack logical architecture [70].

between VMs and thus the view of the guest OS is as a standard machine.

Two options exist to provide network connectivity in OpenStack:

a: PROVIDER NETWORKS

In this scenario, 802.1q tagged VLANS are used between the instances both for management (Controller to/from Compute node(s)) and for the traffic in/out of the environment. It is the role of the physical network infrastructure to provide transport between these elements and thus the physical network must be able to cater for the 802.1q tagging. DHCP services to the Compute Node will services the VMs with IP addresses both for wide-area connectivity and to reach internal resources such as Block Storage and Image Management. In order to provide scalability, DHCP services are often placed on a dedicated Compute Node which hosts just this service and not NOVA functionality.

b: SELF-SERVICE NETWORKS

In the Self-Services Network, an overlay network is created between the various elements as a virtual private network. Whilst this can be created across a Layer 2 network using 802.1q tagging, it is far more usual for this network to be created using VXLAN as a Layer 3 solution. Each VM instance will thus receive an IP address from the DHCP server within the private network and it will be up to the network node in conjunction with the physical network to provide a mapping between the internal IP address and the public IP address on the physical network. Whilst this can be achieved by routing all traffic through the Network Node (remembering that the Network Node is a distributed function which can be associated with any Compute Node), OpenVSwitch can also be used under control of the Neutron L3 service to intercept traffic to/from a VM instance, strip VXLAN tags and re-write Source/Destination IP address as needed.

The connectivity between the internal VM Instance and the real-work network is therefore primarily a NAT function even though the Neutron entity which performs this is referred to as a "router."

2) KUBERNETES

It is important to understand the difference between OpenStack and Kubernetes [71], and for that the difference between a container-style and a virtual machine-style of virtualization must be discussed.

In the virtual machine case, the entire operating system runs in its own environment, fundamentally as if it were built on standard hardware. The Hypervisor provides an abstraction between the Host OS and the Guest OS. In a container solution, the applications are given isolation onto a shared Host OS through their own set of libraries. Shared elements such as networking are common and do not have the same level of isolation as in the hypervisor-based model. However, management of resources between the containers is much more efficient as multiple copies of the operating systems are not required.

In a container-based environment, the applications will require porting to the library-environment of the container host. Docker [72] is a solution for Linux-based containers and has become the de-facto standard. Kubernetes [71] is the management and orchestration solution built on top of Docker to assist with operation of a container environment.

In terms of networking, Docker provides a NAT-ted connection to the outside world using Linux bridges in the underlying OVS. With manipulation of iptables, isolation can be provided between the different containers and it is the provisioning and configuration of these relationships that Kubernetes provides. Linux bridging requires manual configuration on the container host in order to available to Docker and performance can be limited.

Docker also enables use of OpenVSwitch which can then be further enhanced with DPDK to provide higher throughput networking and advanced features such an integration to SDN controllers. Docker makes the interface to the networking layer visible through the Container Networking Interface (CNI) and this has been further enhanced to add the Simplified Networking Overlay Architecture (SONA) which is used as a replacement to Neutron in OpenStack. SONA-CNI [73] when combined with Kubernetes enables the combination of Kuberetes as a Docker solution to control ONOS as an SDN controller and then to manipulate the underlying network on OVS with and without DPDK.

3) CLOUDSTACK

As the second most popular alternative to OpenStack [74], the Apache CloudStack project [75] uses many of the same underlying hypervisor solutions as OpenStack supporting the same mode of virtual machine hosting where an entire software distribution is instantiated as a complete entity in the guest.

In discussing how networking performance may be improved in a CloudStack environment, Vazquez [76] details the configurations required to enable both OVS and DPDK in a CloudStack environment – materially, this presents the same set of issues in terms of performance as in the OpenStack case as the underlying technology is identical.

4) OPENNEBULA

Again similar to both OpenStack and CloudStack in its support for a complete guest, OpenNebula [77] is a further cloud management platform natively using KVM, LXD and AWS Firecracker hypervisor.

In the same manner as OpenStack and CloudStack, OpenNebula when running with a KVM hypervisor may be configured to use OVS and DPDK to enhance network performance over standard Linux network bridges. This technology has only been available in OpenNebula since version 5.10 [78], the release notes stating that the main focus of this version is to support enhanced network performance for NFV applications.

5) PROXMOX

Proxmox [79] supports both containerized (LXC) and full (KVM) virtualization with the same management interface.

As with the other solutions, network performance is limited by the underlying Linux performance – as with OpenStack, CloudStack and OpenNebula, the manner by which this is done is by installing OVS with DPDK support.

With two major virtualization technologies but a common network substrate in term of OVS, it is possible using SONA to create a hybrid solution [80] where the OpenStack and Kubernetes platforms each control the same networking layer via a shared SDN controller. This allows the benefits of each of the virtualization platforms to be used without having to build separate networks.

Figure 5 presents a summary of the available technology enablers and shows their intended use-cases, strengths and weaknesses. This detail should help the network designer select an appropriate solution for their desired architecture.

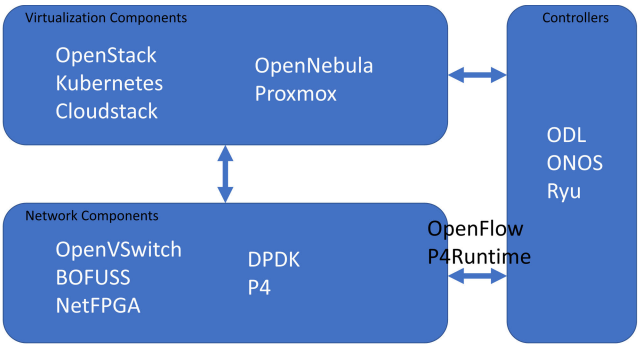

FIGURE 5. Enabler capabilities comparison.

E. PLATFORMS AND ALLIANCES

This group consists of a number of industry bodies who have developed or are developing complete solutions targeting specific areas. In some cases, the group itself is responsible for generating the software; in the other cases, the group is more concerned with creating a solution based on components generated by other bodies.

1) OPEN5GCORE

Based on the 3GPP Release 15 core network specifications [81], Open5GCore [82] is a toolkit available from the Fraunhofer Society's Fraunhofer Institute for Open

Communication Systems (FOKUS). Comprising a full set of 5G core functions, the Open5GCore supports 5G New Radio Standalone (5G NR SA) and network slicing plus benchmarking and performance evaluation tools. Whilst the software is open in that the interfaces to it are defined and documented, the source-code itself is not and access to it is only available under license with Fraunhofer FOKUS to enable prototype development.

To assist with the deployment of testbeds, Fraunhofer provide the FOKUS 5G Playground [83], a 3GPP Rel.16 compliant managed solution. Corici *et al.* [84] discuss how this model enables a core to be constructed within the playground prior to be instantiated on a remote location. They go on to show how the Open5GCore is pivotal in a number of research projects such as 5GENESIS [85] which is concerned with benchmarking KPIs in an end-to-end full-stack 5G solution, 5G-VINNI [65], another KPI-determining project based on industry-vertical specific use-cases and SATis5 [87] a large-scale demonstrator of integration between satellite and terrestrial 5G networks.

2) FREE5GC

Initially, Free5GC [88] was built on the work of nextEPC [89], a 3GPP Release 13 core implementation by National Chiao Tung University (NCTU). There are three current releases of software. Release 1 (January 2019) comprises the basic 4G EPC functionality but with the MME functionality replaced with 5G components AMF and SMF and P/S-GW replaced with a UPF. This also introduced Service Based Interfaces and N4. This limited support to 4G LTE eNB. Release 2 (October 2019) implements additional features to enable support of 5G NR SA gNB and also provides additional network functions on the Service Based Interface whilst moving to a 5G policy function, the PCF instead of the 4G PCRF. Release 2 will also be enhanced with N3IWF and QoS support. Release 3 (March 2020) provides the remaining features of the 3GPP 5GC including application services and support for Network Slicing. The solution may be installed on a variety of topologies from a single server to multiple servers with one-function-per-server. Source code is freely available under the Apache 2.0 License and is in a mixture of C and Golang.

Free5GC is to be seen as a set of building-blocks towards a freely-available and open-source 5G core. As such, despite being a very new development, it is already finding use a number of research projects; Liao *et al.* [90] extend the code to evaluate various Network Slicing schemes for eMBB services based on desired QoS and Tomaszewski *et al.* [91] use the solution to demonstrate a novel integration of 5G and MEC services.

3) OPENAIRINTERFACE

Started as a project of EURECOM [92] in 2014, OpenAirInterface is a development of open-source cellular technologies in three areas; 1) the cellular UE using either pure software emulation or combined with one of a number of open-source Software Defined Radios (SDRs); 2) the cellular xNB using either pure software emulation or combined with one of a number of open-source SDRs; 3) the core network.

Initial work concentrated on the development of a 4G LTE 3GPP Release 8 compliant eNB and associated core network running on commodity x86-based hardware with the Ettus range of SDRs. The direction of the project is controlled by the OpenAir Software Alliance (OSA) comprising a Strategy Board and a Technical Board. Software is released under a range of licenses depending on the usage of any underlying software – the HSS and MME require third-party code meaning that it has to be released with an Apache v2.0 license. Whilst the project originally focused on a total software-stack from xNB to Core, recent developments has seen more priority placed on the xNB components, especially in the area of disaggregated xNB such as the eNB Functional Splits in 4G LTE and the FAPI/nFAPI splits and O-RAN Fronthaul definitions for 5GNR. Compared to developments on the xNB, the Core Network somewhat lags in terms of feature support for 5G. At the moment, the OAI Core Network operates as a full LTE EPC including a split SPGW to support CUPS functionality as an SPGW-C and an SPGW-U. Current work is towards supporting the 3GPP Non-StandAlone (NSA) 5GNR access from gNB in the Option 3X mode, so-called 5G NSA LTE assisted NR connected to existing 4G EPC.

4) OPNFV

Formed in 2014, the Open Platform for Network Function Virtualization [44] is described by Kapadia and Chase [93] as being a ''midstream'' project. OPNFV does not direct or contribute to any upstream open-source project; instead it provides systems-level integration, deployment and testing services of existing open-source components which when combined, create a reference NFV platform based on Enterprise and Service Provider network requirements. New features required in upstream components may be signaled back to the owners of the component for future development and then re-circulated to the OPNFV project for re-integration and testing. The OPNFV project is thus built around three ''Pillars;'' Integration, Testing, New Features.

By utilizing upstream open-source projects, OPNFV is able to provide a set of reference architectures for various use cases based on any combination of functionality in the NFV architecture. Where several open-source projects exist to fulfil the function, OPNFV seeks to provide an integrated and tested deployment. OPNFV operates in a Continuous Integration/Continuous Delivery (CI/CD) model where the work of parallel development teams are merged to a single release with short release cycles. This code is then deployed and tested against the reference models allowing further improvements to be made rapidly. In order to facilitate this rapid development and improvement model, test scenarios are defined and software is automatically deployed and tested as it becomes available. Automatic software installation packages such as Fuel, Apex, JOID and Compass are used to take the upstream projects and deploy. Likewise, a number

of automated test tools are used to both verify the correctness of the installation and its functionality.

Construction and testing of the VNFs which may be deployed within the NVF environment is currently outside-the-scope of the OPNFV project, to some extent due to the lack of standards and commonality between VNF vendors in terms of software packaging and release tools.

One sample VNF which is extensively used within OPNFV is the Clearwater vIMS. Deployment and testing of this VNF has been integrated into the OPNFV project.

5) O-RAN ALLIANCE

The Open Radio Access Network Alliance (O-RAN Alliance) [94] was formed in 2018 by AT&T, China Mobile, Deutsche Telekom, NTT DOCOMO and Orange in order to achieve two main objectives. The first goal is to bring ''cloud-scale'' economics to the 5G Radio Access Network. This means being able to use COTS hardware with software designed in such a way that is deployable automatically and able to scale easily to support an increased user-base. Second, in recognition of the differing and as-yet undefined use-cases of 5G, it should be easy to adapt the RAN components to cope with these rather than having to resort to a complete system rebuild.

The O-RAN Alliance considers this in several areas:

a) Software Defined, AI enabled RAN Intelligent Controller. In order to be able to scale the RAN, human-based interaction will be too slow to adapt. Therefore, the O-RAN Alliance considers how the deployment and operation of the RAN will be managed by AI techniques.

b) RAN Virtualization. A fundamental concept underpinning the Alliance is the virtualization of the RAN at multiple levels allowing scaling to occur. The Alliance is considering existing virtualization techniques and what technological advances may be required to meet the new performance goals.

c) Open Interfaces. The O-RAN Alliance does not stipulate that all software needs to be open-source but it does require that interfaces between functional blocks be open.

d) White Box Hardware. The specification of a hardware RAN platform is being constructed by the Alliance. This will allow any OEM to build an O-RAN Alliance-compliant radio head-end which will interoperate with the other component specifications.

e) Open Source Software. A reference model using open-source software either created by the Alliance or down-streamed from Alliance partners will be made available.

The O-RAN Alliance has developed a Reference Architecture to which members may contribute components and has worked with OPNFV and MCORD to validate the components in reference deployment architectures.

6) TELECOM INFRA PROJECT

Formed in 2016 initially by Facebook, the Telecom Infra Project (TIP) [95] seeks to accelerate the manner by which all mobile telecom infrastructure from the base-station to the core network is built and deployed.

To accomplish this, TIP is split into a number of Project Groups across three broad areas; Access Projects which concentrates on RAN of all kinds including fronthaul; Transport Projects considering non-terrestrial connectivity, optical and packet networks; Core Services Projects currently working on edge services and end-to-end network slicing. Each of these projects combine resources from interested companies who then collaborate on a specific area of development before constructing a representative Proof of Concept in a Community Lab. During Mobile World Congress 2019 in Barcelona, TIP demonstrated a combined Proof of Concept across several key technology areas with contributions from multiple actors.

To-date, all development has concentrated on existing 4G LTE technologies – during late 2019, TIP announced that it would extend the OpenRAN project to include a 5G NR white-box reference architecture. There is significant cross-over between the OpenRAN work in TIP and the work in the O-RAN Alliance although the two bodies remain separate and there has to be some question as to where the balance-of-power lies within the two, with Morris [96] suggesting that the O-RAN Alliance now dominates.

7) CORD/MCORD

The original Central Office Re-architected as a Datacenter (CORD) [97] project was designed around consumer wire-line services in that it provided generalized access switching coupled with a open network operating system, XOS, under control of the ONOS SDN controller and virtualized telecom functions such as the Broadband Network Gateway (BNG) converted to software functions running within an OpenStack and/or Docker environment.

This gives the operator the ability to repurpose or scale the devices quickly in a way not possible with the hardware equivalents leading to greater flexibility in terms of system deployment. CORD was built as a direct spin-off of the original ETSI NfV use-case; that of taking a hardware Broadband Remote Access Server (BRAS) and placing it in generalized compute running with COTS optical switching.

M-CORD is an extension to the original CORD design which takes specific functions in the mobile packet core and distributes these into the same CORD infrastructure. The goal of M-CORD is to produce a useable reference architecture through the demonstration of a Proof on Concept. M-CORD is somewhat similar to OPNFV but the two are not currently linked in any way. In theory, the M-CORD use-case could be deployed within OPNFV and this is suggested in CORD FAQ [98].

F. LIMITATIONS IN EXISTING TECHNOLOGIES AND INITIATIVES

Whilst the softwarization of the packet core has enabled a high degree of flexibility in deployments compared to the monolithic design of previous networks a number of issues arise which around functional placement, the use of cloud technologies as forwarding engines and traffic flow in a softwarized environment.

These issues are discussed below together with details of prototype forwarding solution which combines both software and hardware techniques to balance the needs of flexibility and performance.

1) FUNCTIONAL PLACEMENT

Virtualization of the network function and abstraction of the topology leads to additional complications in that performance, previously relatively easily to analysis by logical study of the elements in the forwarding path, is now more difficult to determine. Virtualization of the network function means that the physical location of the hypervisor supporting the virtual function may not be topologically known to the composed service and there is the risk therefore of sub-optimal placement of the virtual function in the service chain. This lack of visibility is referred to as the functions placement problem and has attracted a great deal of research.

Considering two models of gateway construction, Basta *et al.* [99] details how rather than moving to an all-virtualized or all SDN-decomposed gateway in the data-plane, optimal performance can be achieved with a mixture of both. The authors find that any virtualized devices experience high processing delay for the same function in software as when implemented as decomposed SDN network function and note that a subset of decomposed, hardware-based gateways are always required to meet forwarding requirements. They show a representative example network for the US with a mixture of virtualized and SDN-based gateways.

In attempt to provide a rigorous methodology for optimum function placement in a geographically dispersed network, Luizelli *et al.* [100] proposes a modelling scheme based on heuristic Integer Linear Programming techniques as commonly used in mobile RF allocation schemes using representative SFC traffic. The scale and scope of the problem set used in the paper is not sufficient to be representative of future networking requirements – it was unable to find a solution to a 200 node network in 48 hours of computation given a maximum end-to-end latency of 90ms.

2) PACKET FORWARDING PERFORMANCE

Assuming that the issue of optimal function placement can be overcome or at least better understood as the packet core scales to support 5G and beyond, concerns over the packet forwarding capabilities of the elements in the core remain. Raumer *et al.* [102], Kawashima *et al.* [103] and Pitaev *et al.* [104] each discuss the issue of traffic performance and detail potential bottlenecks in current solutions. Raumer *et al.* [102] note that there is little detailed work in this area and go on to show significant performance degradation through chained links between forwarding elements built using both Open vSwitch (OVS) and Linux namespaces. Of concern is how the throughput in an SFC network comprising only 10 links drops to less than 0.3Mpps based on an offered load of 2Mpps. Latency increases from $10^1\mu$s to nearly $10^4\mu$s for the same scenario.

Similar results and issues are noted by Kawashima *et al.* [103] who extend the performance testing to include other data-plane technologies such as OVS with Data Plane Development Kit (DPDK) and netmap-based solutions. Each of these suffer with similarly high levels of throughput issue. Kawashima *et al.* [103] note that the only currently acceptable solution to achieving high throughput from the wire to a VM-based VNF is the use of Single Root Input/Output Virtualization (SR-IOV).

When considering how traffic may be chained through several UPFs, Ge *et al.* [105] show how not only is East-West traffic impacted in a pure-software solution but how performance varies according to the location of the VNF inside or outside the same hypervisor. Mitigation techniques involved transition of traffic across external hardware switches are also discussed.

Pitaev *et al.* [104], having documented the points in the virtualized system architecture where bottlenecks can occur, extend the performance measurement to include the FD.io Vector Packet Processing (VPP) data-plane which is comparable in performance with OVS-DPDK with slightly higher performance in terms of both throughput and latency. Both OVS-DPDK and FD.io are based on DPDK technologies so this is not a surprising result The results mirror those of Kawashima *et al.* [103] and both agree that at present, in terms of wire-to-VM, the only viable option is SR-IOV for maximum throughput and minimum latency.

Common to all three performance studies is the lack of linearity in all solutions except SR-IOV. Native OVS is shown to tail-off at around 2Gbit/s; FD.io and OVS-DPDK around 7Gbit/s whereas SR-IOV remains linear to around 15Gbit/s each based on a 20Gbit/s Network Interface Card (NIC). This is a worrying aspect which is difficult to account for in the design and operation of any future network especially given that SR-IOV is, as stated by Kawashima *et al.* [103], hardware inflexible and processor dependent. SR-IOV relies on a 1:n mapping between the physical NIC and a virtual machine instance through specific configuration in the hypervisor – the n indicates that, dependent on the NIC card capabilities, a single card may be associated with multiple VMs. The inflexibility comes from this hardware mapping – moving the VM to another hypervisor requires that the mapping between the NIC card on the new hypervisor and the VM be manually configured. This is similar to techniques such as Peripheral Component Interconnect (PCI) Passthrough used to enable Graphics Processing Units (GPUs) to be directly addressed from a VM for optimum graphics performance.

Whilst no longer the case, SR-IOV was originally an Intel-only solution for both NIC and CPU. It does still require support in the Intel CPU but is no longer limited to Intel NICs alone. However, the lack of flexibility runs counter to the flexibility promised by NFV.

Lab results taken over the past few at the 5G Innovation Centre and presented later in this report agree with and verify these findings. As well as individual performance characteristics, Pitaev *et al.* [104] look at the performance of VMs

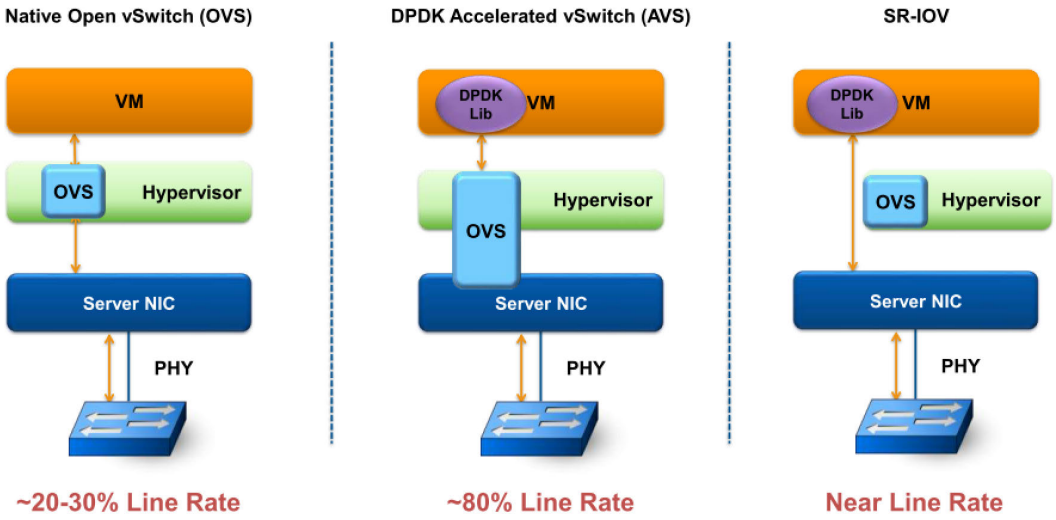

FIGURE 6. Connectivity comparisons [101].

running concurrently in the same hypervisor connected to the physical infrastructure by a variety of technologies, namely OVS-DPDK, FD.io VPP and SR-IOV. There are several important conclusions to be drawn from this work including implications for multi-socket systems. Pitaev *et al.* [104] show that as before, the only technology which is consistently able to deliver a linear throughput to individual VMs is SR-IOV and that for both OVS-DPDK and FD.io VPP, performance decreases after reaching an initial plateau at around 5 parallel VMs for OVS-DPDK and around 6 for FD.io VPP. The authors then go on to state that the cross-socket issue - whereby a VM becomes associated with a network service on a NIC which is on a different PCI bus due to the NUMA architecture - is masked in both OVS-DPDK and FD.io because the underlying technology of DPDK has become saturated with network traffic. Figure 6 summarizes the termination performance of different strategies.

The performance papers also make an assumption that either the VM itself is the sink (i.e. the traffic flow occurs between the end user device and an application within the VM) or that the data-forwarding elements are wholly composed of functions such as OVS or FD.io. This is true in the case of a cloud service where data-forwarding across a chain may occur prior to termination in the host application, but in the case of NFV, the forwarding function itself which passes across the user-plane of the VM between virtual NICs must be addressed.

There are two aspects to be considered in this regard in relation to mobile packet core networks. First the performance of the forwarding element will be negatively affected by processing of the TEID in user-space of the VNF. This is the case where a standard LTE S/P Gateway which has similar functionality to the 5G UPF is virtualized and run as a software element [106]. Deploying the TEID components in an accelerated data-plane, for example by programming TEID forwarding directly into DPDK as the authors have done, helps, but as shown still hits the DPDK performance barrier discussed previously. Work by Pinczel *et al.* [107] on improving performance for the packet forwarding of the Click modular router compares user-space forwarding with accelerated modes (here based on netmap which is directly comparable to DPDK) – the accelerated modes show up to a 10x increase in packet-per-second throughput. The implication therefore is that any processing of packet forwarding carried out in the user-plane will be severely impacted. Second, there is the question of whether the data-plane technology can support the throughput required in order to both deliver data through the VNF and on to the next VNF. In the case of a Service Function Chained solution comprising several UPFs each composed of one-or-more VNFs, there could be many instances of data-forwarding in a software data-plane and across the user-plane in the VNF which would comprise the chain.

3) EAST-WEST TRAFFIC PERFORMANCE IN A VIRTUALIZED ENVIRONMENT

It is important to note that the original use-case for virtualization was not as a data-forwarding path but as an application end-point. Also, all the data-plane acceleration techniques so far discussed operate in the North/South domain, that is NIC to/from VNF. In the case of a Service Function Chain composed of several parallel VNFs, elements of both North/South and East/West traffic will be required.

In this context, East/West is defined as traffic between two VNFs where their hosting VMs reside on the same hypervisor and there is a hypervisor-specific data-plane in operation between the two VNFs. Where traffic has to leave the hypervisor to reach a VNF on another hypervisor, this can be seen as two North/South transitions and a physical network hop.

The problem of East/West performance appears not to have been greatly addressed in research as it has not been

seen as a general use-case for virtualization and the current solutions, DPDK and SR-IOV all address the physical-to-virtual path, not the virtual-to-virtual path. Some clues as to potential performance degradation in the East/West path are discussed by Hemminger [108] but these are in ideal conditions using dedicated Linux bridges. Even here, the highest VM-to-VM traffic achievable without offload techniques was 7Gbit/s - Hemminger [108] suggests using an SR-IOV based NIC associated to both VMs and transitioning traffic through this device where 17Gbit/s was achieved.

With an attempt for finding an alternate data-plane in the hypervisor, Mwangama and Ventura [109] propose replacing both OVS and OVS-DPDK with a kernel-based module, eXpress Data Path (XDP), which handles all data-plane traffic in the hypervisor, exposing elements via IOModules to the VMs. In this respect, it is similar to SR-IOV in that the known issues of OVS are bypassed, yet it retains commonality between the VM network interfaces in a manner similar to OVS. As a replacement for OVS, XDP requires a complex, non-standard integration with virtualization technologies such as OpenStack. Initial performance results with XDP show that there is little difference in performance whether the traffic is North/South or East/West. However, tested traffic levels at 100Mbit/s are very low and the results are very hard to interpret. The solution would also still suffer with the need to cross the user-plane in the UPF VNF which is not discussed at all in the paper. Significant work would be required to re-create the environment and test at more representative speeds although the promise of SR-IOV like features without the hardware implications is attractive.

Each approach at solving the performance issue in the Virtual Machine/Hypervisor realm has so-far been limited. An alternative approach offers a different set of networking options in the form of container-based virtualization. The predominant method of network connectivity in containers is the Linux Name Space whereby an entire copy of the host networking stack is made available to the virtual machine. Popular container solutions such as Docker are also able to make use of other data-planes such as Linux bridge or OVS, but Amaral *et al.* [110] show how near bare-metal performance may be achieved to a container-based application using the Name Space construct in both North/South and East/West configurations. Amaral *et al.* [110] also document performance for the same transit paths for both Linux Bridge and OVS showing significant performance issue for the latter two cases in the East/West path as expected.

To illustrate and determine the difference in performance by Linux Name Spaces and OVS when used as a Service Function Chain, Livi *et al.* [111] construct representative chains of up to 20 Name Spaces and OVS entities, showing how the Name Space chain consistently out-performs OVS. Whilst at face-value, the use of containers in the forwarding plane would seem to offer an answer to the virtualization networking performance ''tax,'' this is only true if the VNF were able to use the host networking directly as is the case with a VNF hosting an end application. Again, where the VNF is itself part of the data-forwarding path, the same issue as with traditional virtualization remains – the packet needs to enter the user-plane in the VNF, be treated by the UPF function(s) and then passed back to the data-plane. Unlike in traditional virtualization, in the case of Name Spaces, the data-forwarding path is shared by the host and the virtual machine – one could consider implementing kernel-based switching in the VNF as a means of improving data throughput, but in the case of a container-based solution, this data-path is not ''owned'' by the VNF but by the hosting system. As each VNF inherits a copy of the single host data-plane, functionality cannot be unique to each Name Space.

4) BALANCING PERFORMANCE AND FLEXIBILITY

The goal of flexibility in the 5G core is met with the application of softwarization to the architecture, but the question remains as to whether this flexibility comes at the cost of performance. Condoluci and Mahmoodi [112] discuss how the benefits of softwarization provide flexibility in terms of speed of innovation and scalability, but do not consider any performance issues, indicating that this remains an open area for future research.

The issue for potential performance degradation in softwarized environments was discussed in 2015 in an ETSI report [113] and generic use-cases were described. However, this report pre-dates many of the techniques detailed in this paper. Additionally, it does not attempt to measure or otherwise describe any potential impact. A further descriptive document from ETSI in 2017 [114] details a range of use-cases and mentions that performance is impacted but provides no measurements and makes no recommendations as to how to resolve any issues.

Despite much performance measurement and documentation of the issues of network performance in virtual environments including lessons learnt within the 5G Innovation Centre [42], the issue of function placement and associated network performance especially for VNFs hosting UPFs remains an open research question.

This combination of SDN, NFV, compute virtualization, Service Function Chaining and multiple network paths makes service analysis very complex as there are many ''moving parts.'' Nam *et al.* [115] present an interesting project, Probius, which aims to take an holistic view of the entire chain by collecting and analyzing a range of performance data from points in the service chain and alerting to abnormal situations in an attempt to address performance uncertainty.

Table 2 summarizes the current situation. In the case of the bare-metal solution, this deployment option offers the best in terms of both North/South and East/West (which is essentially two North/South connections between adjacent systems), but lacks any support for SDN control, SFC support and none of the benefits of virtualization flexibility. This represents the current bench-mark solution of a network function which is typically a monolithic, for-purpose network element.

Due to the lack of flexibility, the Bare-Metal solution is immediately discounted as it cannot offer the basic

TABLE 2. Technology comparison.

	Flexibility of Deployment	SDN Control	SFC Integration	North/South Performance	East/West Performance
Bare-Metal	Fixed Config	None	None	Best	Best
Virtualization with OVS	Highly Flexible	Yes	Yes	Poor	Poor
Virtualization with DPDK	Highly Flexible	Yes	Yes	Good	Poor
Virtualization with SR-IOV	Hard-coded NIC Config	Limited	Yes	Good	Good (via shared NIC)

requirements imagined in NFV. For similar reasons, the SR-IOV case where an SR-IOV-enabled NIC is hard-coded to the virtual instance the traffic is destined for, is also discounted. It is limited a number of proprietary Intel and Mellanox NICs.

However, literature suggests that in certain configurations where a single SR-IOV-enabled NIC is shared between two virtual machines, good performance in the East/West direction can be achieved.

Options using OVS offer maximum flexibility and can be integrated with SDN and SFC solutions. Un-accelerated OVS impacts performance compared to bare-metal in both the North/South and East/West directions – enabling DPDK allows an improvement in the North/South direction but does not address the East/West case. No current SDN/NFV solution is able to offer the promised flexibility of deployment and operation whilst maintaining or improving the performance of the infrastructure. In the single data-forwarding case such as in today's mobile packet core networks, performance is compromised when moving to a virtualized solution. In future developments which use multiple UPFs chained with SFC, this loss of performance will be multiplied at each UPF boundary. New techniques such as a different split of hardware and software functions in the data-plane should be developed such that optimum switching performance can be achieved. Stateful switching will help alleviate the controller placement problem which will be more prevalent in highly geographically-distributed networks.

In describing their LTE network in a PC, Nikaein *et al.* [116] describe ''performance'' as the placement of the entire core at the eNB location, not in distributed environment. In subsequent demonstrations, Nikaein *et al.* [117] show OpenStack being used to provide the EPC functionality but the loading on the EPC from a single eNB is not sufficient to impact performance and no measurements of the system under stress are made.

One of the first fully-softwarized 5G cores built on 4G EPC components modified to operate in a pre-standard 5G-like manner was presented by Lake *et al.* [42] and issues were seen with packet performance degradation initially pointing at known issues in OpenVSwitch. Further, it was noted that existing research into this issue had concentrated on the VNF as a terminating, not a transiting application [118]. Two further papers, by Oh *et al.* [119] and Ge *et al.* [105] discuss the degree of the impact by providing measurements of the loading in different scenarios. Oh *et al.* show how network throughput to a virtualized softwarized core is degraded by up to a factor of 10x and how the peak RTT is also increased. Because direct comparison is made between a softwarized core and a softwarized and virtualized core, the authors show that the major impact is in the manner of the virtualization as stable performance is achieved in the non-virtualized solution. Ge *et al.* further show that in specific cases where traffic is being passed between two-or-more UPFs in a virtualized environment, performance is greatly impacted both in terms of throughput and RTT.

5) HYBRID SOFTWARE/HARDWARE SOLUTIONS

It appears therefore that a balance needs to be struck between full flexibility and desired performance and in designing a fully softwarized, fully virtualized 5G Core, the interaction of every element becomes important and must be considered. In some cases, for example where the desire is for low-latency, high-throughput traffic, it may not be possible with current virtualized solutions to deliver an acceptable outcome. In the context of network slicing, this could mean that certain classes of traffic be handled by non-virtualized slices.

Tools such as the NetFPGA and data-plane programming languages including P4 make possible the realization of such combined hardware/software solutions such as the prototype of a hardware-accelerated, virtualized packet core built at the 5GIC by and shown in Figure 7. and show its interaction on an SFC flow for achieving maximum performance.

Whilst the software forwarding plane remains in the virtual environment and therefore running in software, the programming of the UPFs from the 5G control-plane is also sent to the network programming layer which controls an FPGA-based hardware switch. A decision can be taken at network forwarding time as to whether the packet should transition the switch to the software forwarding elements or be ''cut-through'' in the hardware switch. Any changes of header, for example SFC label or TEID, is programmed into the switch by the Data Plane SDN Controller. This keeps the softwarization of the 5G core in place whilst using advanced SDN and hardware forwarding techniques to ensure optimum performance where this is required.

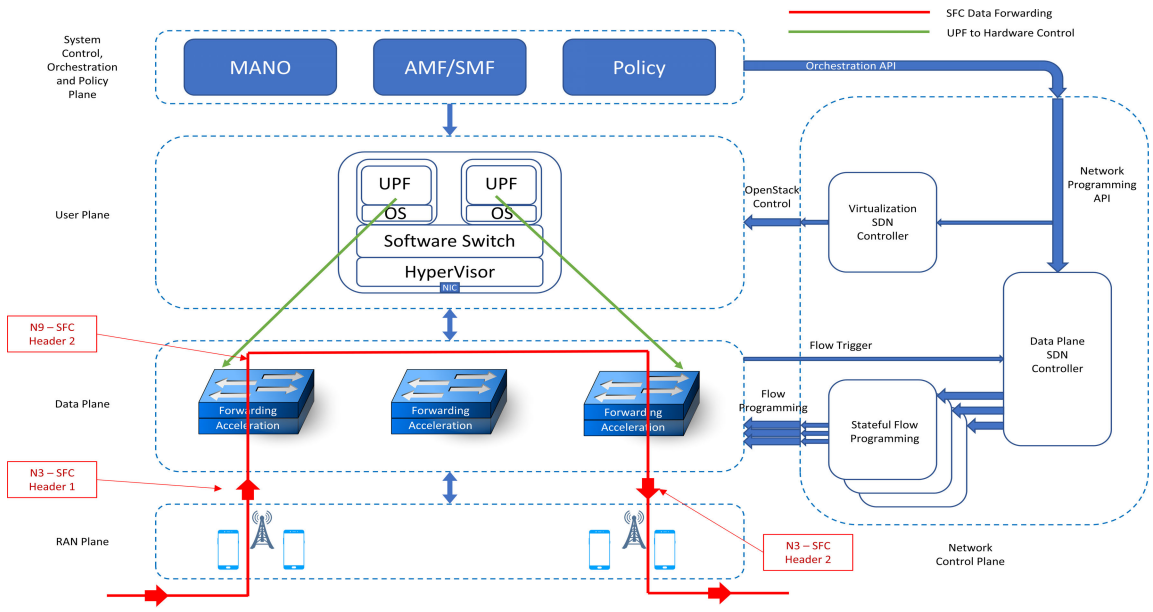

FIGURE 7. Hardware-accelerated virtualized EPC.

As a hardware solution based on both open-source hardware and software, the NetFPGA seeks to address both the performance and open-source requirements, but the amount of custom programming required is a limitation as it will make each implementation individual rather than standardized. On the other hand, P4 and P4Runtime seek to maintain the hardware and performance elements of the NetFPGA project whilst proving a standardized, softwarized programming interface through P4Runtime. There is one caveat – P4 allows for closed-source modules to be inserted into the P4 pipeline and would therefore turn an open-source implementation into a close-source one.

Moving forward, P4 and P4Runtime appear to offer the promise of softwarization of the data-plane in both a performant and open method.

IV. STANDARDIZATION LANDSCAPE

Supporting the development of both the RAN and Packet Core in 5G and beyond are an array of SDOs. At first glance, the number and scope of the these groups can seem daunting – organizational differences in their operations, attendance and funding can also mean that at times there appear to be overlap between them and at others vastly diverging aims. This chapter not only surveys and summarizes each of the relevant SDOs to the development of 5G and future packet-core technologies, but also offers a commentary on the operation and basic rules-of-engagement of each to assist the reader in determining how best to contribute to the groups.

A. ITU IMT-2020

During 2012, ITU embarked on a project to consider the future uses for mobile communications systems beyond the current deployed 3G and 4G technologies. This set of work culminated in the publication of ITU-R M.2083.0 [120]. Whilst the ITU had the ability to consider the future growth of mobile traffic, it did not have the remit to develop the technical standards in support of that growth and whilst the document makes predictions for future network traffic, coverage and use-cases, it does not present any direction towards technical solutions to enable these outcomes.

IMT-2020 produced a taxonomy of connectivity which has since become the guiding force behind all 5G network design:

- Enhanced Mobile Broadband (eMBB); an extension of existing services on mobile 4G networks requiring high user-experienced data-rates and efficient geographical coverage with support for mobility.
- Ultra-reliable and Low-Latency Communications (urLLC); a new service model where outcomes in terms of persistency of connections, mobility and low-latency are defined but without the need for wide-scale coverage or high data-rates
- Massive Machine Type Communications (MMTC); connectivity at low data-rates, without mobility, less important latency requirements but the ability to support many such end-points.

Once published, the IMT-2020 goals formed the design-goals around which institutions were able to understand the requirements of both the Radio Access Network and the Core Network in 5G. Work to enhance and modify the IMT-2020 system specifications is handled in ITU-R by Working Party 5D [121], a WP that remains active during the technical development and standardization phase. IMT-2020 is a sub-group of the wider ITU-T Study Group 13 [122], chartered with considering future networking technologies in the 2017-2020 Study Period.

B. ITU FG NET-2030

Focus Group Technologies for Network 2030 [123] was formed in July 2018 with a charter to consider network development to 2030 and beyond. It was formed as a further sub-group of the existing SG13 but unlike IMT-2020 which will close at the end of the 2017-2020 Study Period, FG NET-2030 is designed to continue beyond this period, specifically considering the type of network requirements which will be prevalent in the 2025 – 2035 timeframe.

The Terms of Reference of FG NET-2030 discusses a number of technologies already seen in current IMT-2020/5G requirements but imagines how the demands made of these technologies increase to a level beyond the scope of IMT-2020. For example, FG NET-2030 lists three areas of critical interest beyond 5G but which are direct evolutions of the current state-of-the-art:

(1) Astronomical amount of connections beyond the limitation of current and near future networks including 5G/IMT-2020. Current network design typically has a low number of connections compared to future predictions for connected devices. Consumer networks and the components that comprise them were designed at a time when the number of devices was very-much-less than one-per head of population. Cisco VNI [124] show that in 2016 there was a global total of ∼7 Billion connections – this is predicted to reach over 16 Billion connections in 2021. This massive increase in connections will challenge current networking technologies.

(2) Very-high throughput to support explosive bandwidth-intensive future applications beyond the limitation of current and near future networks including 5G/IMT-2020. Coupled with this increase in the number of connected devices is a corresponding increase in the use of high-bandwidth applications such as video. Figures for 2016 show that 70.3 Exabytes of video per month were delivered – predictions for 2021 show this increasing to 227.6 Exabytes [124, p. 2].

(3) Super-ultra-low latency networking, with deterministic guarantee beyond the limitation of current and near future networks including 5G/IMT-2020. Even more challenging is the rise in the needs for connected devices not currently wide-spread in their use. Ranging from simple sensors to complete factory systems, holographic communications and haptic control used in remote surgery, the need to be able to deliver against a defined outcome will be of vital importance and not something today's networks are readily capable of. Current networks are also more designed around long-lived, large packet flows – many of the newly-emerging use-cases utilize small amounts of data carried against strict Service Level Agreements. It needs to be determined whether the current suite of Internet Protocols are best-placed for transporting such requirements.

(4) Trustable network infrastructure. Many aspects of today's networking technologies in terms of security and trust were provided as overlays or add-ons. The level of trust that can be placed in the transport infrastructure in the future will be a deliverable in a Service Level Agreement, but the current solutions do not allow the requirement to be so decomposed. In addition, where lawful need exists to monitor or intercept traffic, this must also be part of the network infrastructure – current ''Lawful Intercept'' [125] mechanisms in use on mobile networks often result in sub-optimal routing of traffic which is detrimental to all users. Overlay trust and security mechanisms such as IPSec [126] also burden communication networks with additional traffic and overhead; this is incompatible with the need to increase the efficiency of traffic handling due to explosive growth.

(5) Human safety and privacy centric reliable networking mechanism. Parallel to the trust and security needs are the requirements to preserve personal privacy and ensure safety. FG NET-2030 will consider how these requirements can also be built into future networks.

C. 3GPP

The organization responsible for delivering standards to which components and systems can be built, the 3GPP [127] was chartered by seven ''Organizational Partners'' representing the development communities of seven different regions (ARIB [128], ATIS [129], CCSA [130], ETSI [131], TSDSI [132], TTA [133], TTC [134]). The membership is made up of telecommunication operators, academic institutions and equipment manufacturers either directly or through their regional partner standards body.

The term ''5G'' relates to the 5th generation of mobile technology which 3GPP has standardized and hence is analogous to ITU's IMT-2020 vision. There is therefore a symbiotic relationship between ITU and 3GPP – ITU feeds high-level vision to 3GPP which then considers the method by which these requirements may be met, contributing its own output back to the ITU WP5D owning the IMT-2020 vision. ITU will then decide whether the contributions made meet the requirements before progressing. As 5G is an area of active development, 3GPP describes the situation with 5G as a ''bit of a chicken and egg scenario'' [135].

3GPP works to defined time-lines – standards are completed according to a timetable set out by agreement at the start of a standardization project. Standards are declared complete against the timetable, not functionality, a distinction which becomes important when considering relationships with other SDOs such as IETF. From a core network perspective, there is little difference between the two 5G Phases defined by 3GPP. Phase 1, corresponding to Release 15 [80], brings in the concepts of Network Slicing, building on the Control and User Plane Separation introduced to LTE in Release 14 and enables 5G New Radio (5GNR) to be deployed in both standalone and non-standalone modes. Release 16 [136] expands radio connectivity to include New Radio in Unlicensed spectrum, V2X and non-terrestrial connectivity but does not introduce new core network functionality.

To-date, the network functionality in the 5G core in terms of network components and protocols in the user plane

remain the same as for LTE, albeit with different interface names to allow for multiple points of attachment, for example in the case of a sliced network. Release 15 completed in June 2019, changes having been frozen in March 2019. Release 16 will complete in June 2020 with a freeze date in March 2020. It is therefore unlikely that any radically new core network changes will be made in the Release 16 timeframe. Release 17 has started the initial discussion phase as of January 2020 and will be completed in three phases – phase 1 in December 2020, phase 2 in December 2021, and phase 3 in December 2022.

In discussing the current state of the core network technical work, Guttman [137] shows how aspects of the Next Generation Network are out-of-scope of current 3GPP release processes.

The latest 3GPP release, Rel.17, [135] already includes a number of study items considering interaction of the core network and it is therefore likely that with the major 5G RAN components complete, new standardization work will concentrate on the aspects of the core.

D. ETSI

ETSI [131] was chartered in 1989 with developing a pan-European digital cellular telephony system when the *Groupe Spécial Mobile* committee of the CEPT was transferred to them [138].

Within 2 years, ETSI had standardized the first digital mobile system in the world, now known as the Global System for Mobile Communications (GSM) and the first calls were made in 1991 in Finland. Realization of the entire capabilities with enhancements for data support through General Packet Radio System (GPRS) [139] and roaming between operators followed swiftly.

As it became obvious that the need for data and international roaming outstripped the capabilities of these 2G systems, ETSI formed 3GPP in order to develop a replacement. As such, ETSI can be seen as the fore-runner of today's 3GPP organization and many of the administrative functions and operations of the two organizations remain similar. In terms of development of the 5G Core, ETSI is both a constituent part of and an external contributor to 3GPP.

Of particular note are four groups highly influential to packet core development run by ETSI:

1) NETWORK FUNCTION VIRTUALIZATION INDUSTRY STANDARDS GROUP (ETSI NFV ISG)[140].

Founded in 2012 and now working on Phase 4 definitions, this ISG is chartered with developing standards around the Network Function Virtualization elements of the Service Provider core. This is complementary to SDN – ETSI details that NFV ''offers the capability to manage and orchestrate the virtualization of resources for the provisioning of network functions and their composition into higher-layer network services.''

2) MULTI-ACCESS EDGE COMPUTING INDUSTRY STANDARDS GROUP (ETSI MEC ISG) [141].

Originally termed Mobile Edge Computing, ETSI's charter for MEC states that ''MEC offers application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the network.'' This work acknowledges the core aggregation bandwidth issues and latency which is experienced by distant-placed applications and seeks to work around these issues by placing compute closer to the client.

3) NEXT GENERATION PROTOCOL INDUSTRY STANDARDS GROUP (ETS NGP ISG) [142].

Perhaps more long-term and research-focused than many of the other ISGs, the Next Generation Protocol group is studying potential replacements for IP in the packet core of future mobile networks. It remains to be seen how this work will progress – whilst ETSI has strong ties to 3GPP, both are really ''consumers'' of pre-existing networking protocols in the most part developed by IETF. Parallel work in the IETF and IRTF should be considered.

4) ZERO-TOUCH NETWORK AND SERVICE MANAGEMENT INDUSTRY STANDARDS GROUP (ETSI ZSM ISG) [143].

ZSM ties together the management and provisioning of NFV and SDN entities such that delivery, deployment, configuration, assurance, and optimization of resources within the environment will be handled automatically against SLAs. For example, provision of a MEC resource would be handled when demand required and would take no manual intervention. Formed in late 2018, work has been swift in creating a multi-vendor architecture for zero-touch network configuration and service management.

E. IETF

Run under the auspices of the Internet Society (ISOC) [144], the IETF [145] exists to provide standardization that enables the Internet to ''work better.''

In practical terms, this means that the IETF is concerned with the interoperability and engineering of components that form the core of the Internet. This includes aspects such as ensuring that protocol design is coherent and inter-operable between vendor implementations, network management is able to adequately address components and other facilities such as security is understood.

The IETF does not produce architectural-level documents – the design and construction of the components is left to the manufacturers but the interfaces to those pieces are expected to conform to the IETF agreements. Because this is a ''build-first, standardize later'' environment, many protocols start as proposals from individuals which are then refined in a series of Working Groups, each WG being a member of an Area of technology to which it is relevant. Overseeing the entire ensemble is the Internet Architecture Board, the IAB [146].

The key mantra of the IETF is "rough consensus and running code." This emphasizes that the IETF will not work on a subject to the point where everyone agrees and that operational systems are preferred over purely theoretical ones. Likewise, the IETF focus is on the individual, not the country or employer.

Despite the strong use of Internet technologies in current mobile systems, the mobile architecture has traditionally not been an area that the IETF has concentrated on; the relationship has been more of the mobile world being a consumer of IETF protocols. A case in-point is the use of the Generic Tunneling Protocol (GTP) in the mobile packet core – until recently, GTP was only described by a 3GPP definition [147] as it was simply a use of an UDP/IP datagram. It has been added to the IETF through draft-hmm-dmm-5g-uplane-analysis-01 [148] as of late 2019.

Of particular relevance to 5G and future mobile packet core are the following IETF groups:

1) DISTRIBUTED MOBILITY MANAGEMENT (DMM) [149]

DMM was originally chartered in 2013 and was mostly concerned with the mobility requirements of 802.11 WiFi networks. Cellular networks carrying data were seen more as an extension of the fixed Internet.

Over time, the focus of the group has changed and today it is the primary group in the IETF considering cellular-based mobility data protocols and support, specifically around solving the data anchoring issue in current the packet core.

draft-ietf-dmm-distributed-mobility-anchoring [150] presents solutions around Client and Network based mobility anchor points, allowing IP sessions to survive a mobility event on either side of the connection.

draft-ietf-dmm-fpc-cpdp [151] is an active draft discussing standardization of the Protocol for Forwarding Policy Configuration (PFPC) which provides separation between the control-plane and data-planes in a packet core. Note, this should NOT be confused with the 3GPP Packet Forwarding Control Protocol (PFCP) TS 29.244 [152] which provides forwarding programming between the control-plane and user-plane in a 4G Control and User Plane Separation (CUPS) build and in 5G Core and runs over the 4G Sx and 5G N4 interfaces

draft-ietf-dmm-5g-uplane-analysis [153]. From an active research perspective, this is the most interesting active Draft as it is concerned with analyzing and recommending potential protocols to replace the current GTP mechanism used in both 4G EPC and 5G Core. It is also the place where the current protocols are described in detail and thus forms the system interface description to which any component designed to fit in a 4G EPC or 5G UPF are expected to comply to. Following analysis of the protocols, this document is designed to form a set of recommendations to 3GPP's "TSG Core Network and Terminals WG4" (CT4) [154] in terms of appropriate packet protocols for use in the User Plane through the Internet Draft bogineni-dmm-optimized-mobile-user-plane [155]. To-date, four potential protocols have been recommended all of which have discussed relevance to 5G packet cores:

- SRv6 – Segment Routing as a replacement for GTP in the mobile packet core. This is the most advanced of the proposals with a published Internet Draft ietf-dmm-srv6-mobile-uplane [156].
- LISP – Locator and ID Separator maps locally-significant Endpoint Identifiers to globally-routable Routing Locators (RLOC) and therefore allow a separation between the mobility needs and fixed routing. A discussion on implementation in a mobile network has been produced by Farinacci [157] but work on this solution is not complete.
- ILA – Identifier/Locator Addressing is similar to LISP but is more embryonic. A discussion on the protocol and application to mobile packet core is provided by Herbert [158].
- Hybrid ICN – Hybrid Information Centric Networking uses the Producer/Consumer model of traditional ICN and maps this to routable IPv6 addresses which are presented as IDs rather than globally routable addresses. Thus, IPv6 networks can carry the hICN traffic but only hICN nodes can action the mapping between Producers and Consumers. This approach is discussed by Auge [159].

2) 5G ASPECTS OF NEXT GENERATION INTERNET PROTOCOLS (5GANGIP) [160]

This proposed working group is considering new protocols which could replace IP within the 5G Core. It is taking a neutral view of the current possible protocols and building a list of potential candidates. It is important that the work of this group is followed as it will have a direct impact of the softwarization of the 5G Core. In the User Plane, the elements in the data path will need to be able to support non-IP packets – many software stacks use devices such as offload of packet processing to increase throughput and these techniques are limited to support IP only. In the Control Plane, the instructions sent to the UPF will need to be able to carry network-level addressing in the format used by these new protocols.

3) NETWORK SLICING (NETSLICING) [161]

As IETF is primarily concerned with standardizing existing solutions, this proto-WG has found it difficult to coalesce around a definition of a "Network Slice" and discussion continues in order to be able to create a Charter that a full Working Group can use. 5G Core softwarization has concentrated to-date around the 3GPP definition of Network Slicing as discussed earlier in the document. As this IETF group works to build different definitions of Network Slicing, work will be required to support these in the 5G softwarized solutions.

4) RELIABLE AND AVAILABLE WIRELESS (RAW) [162]

Scheduled for an Information BoF in November 2019 with potentially WG-forming BoF in March 2020, this group aims

to discuss how protocol enhancements may be used to provide deterministic transport over wireless links. At present, the scope is limited to IEEE 802.11-based solutions but this may grow to include any form of wireless-based connectivity such as cellular. Future developments from this group will impact 5G softwarization it two ways: first, it may be that the softwarized core needs to be expanded from supporting current cellular-only components to include other elements of connectivity; second, the softwarized core will need to be able carry both the User and Control Plane messages of non-3GPP protocols, for example a for-purpose IoT protocol with a non-IP packet and addressing format.

Supporting the development of both the RAN and Packet Core in 5G and beyond are an array of Standards Development Organizations. At first glance, the number and scope of the various SDOs can seem daunting – organizational differences in their operations, attendance and funding can also mean that there at times appears to be overlap between them and at others vastly diverging aims. This chapter not only surveys and summarizes each of the relevant SDOs to the development of 5G and future packet-core technologies, but also offers a commentary on the operation and basic rules-of-engagement of each to assist the reader in determining how best to contribute to the groups.

F. IRTF

A sister organization to the IETF and also chartered under the ISOC, the Internet Research Task Force (IRTF) [163] takes a long-term view of a wide range of future networking requirements and forms small Research Groups (RGs) to study these developments. In terms of research into future mobile technologies, there are several IRTF RGs which are of interest:

1) COMPUTING IN THE NETWORK PROPOSED RESEARCH GROUP (COINRG) [164]

This proposed RG seeks to consider the impact on network architectures of the availability of compute resources within the infrastructure both as an extension of and replacement for traditional data centers. This is complementary to the work on Edge computing happening across the industry. In 5G, Multi-Access Edge Computing is used to distribute content and applications to the topologically most effective point. Therefore, the *coinrg* can be seen as a super-set of all distributed computing scenarios and MEC as a specific case of distributed compute for the 5G environment. As *coinrg* develops into first a Research Group and eventually an IETF Working Group, the experience of MEC in the 5G softwarized core will become a use-case of a wider definition.

2) INFORMATION-CENTRIC NETWORKING (ICNRG) [165]

Started in 2012, ICNRG studies all forms of Information Centric Networking including NetInf and Named Data Networking (NDN). It is perhaps the most mature of the Research Groups and possibly one of the most practical as it proposes replacements or additions to existing Internet Protocols. The RG is currently concerned with research into naming schemes and data storage and retention policies. As ICN includes the ability to cache content [166] there are some parallels with both the ETSI MEC work and the *coinrg*. ICN also has the potential to be a replacement for IP and therefore support from the UE through the 5G Core could be required. This has an impact on the core in the same manner as the earlier discussion on the IETF *5gangip* Working Group as the softwarized elements would need to support a non-IP protocol. Proposals to use the path-selection inherent in ICN to replace the current anchored IP sessions in the packet core have been made [159].

3) PATH AWARE NETWORKING RG (PANRG) [167]

Current Internet architectures hide much of the information about the transport layer from the endpoints which typically assume that their connectivity to the network provides an adequate description of the entire path.

PAN RG recognizes that current Internet connectivity, especially in the case of mobile connectivity, cannot be assumed to provide a constant connection profile of this sort. This recently chartered RG has three main aims in its research:

- To study the communication and discovery of information about the properties of a path on local networks and in internetworks, exploration of trust and risk models associated with this information, and algorithms for path selection at endpoints based on this information.
- To design algorithms for making transport-layer scheduling decisions based on information about path properties.
- To design algorithms for reconciling path selection at endpoints with widely deployed routing protocols and network operations best practices.

Much discussion at the moment in *panrg* is around the need to collect connectivity-specific metadata, for example radio conditions, multi-point connections, etc. Softwarization in 5G exposes this metadata and therefore can be used to educate the path information. In *panrg* work has already begun to describe paths across different access media using Path Properties - Enghardt and Kraehenbuehl [168] shows how these may be applied to multiple access technologies including 5G.

Table 3 shows the areas of softwarization each of the SDOs concentrates on.

V. KEY OBSERVATIONS AND OPEN ISSUES

In considering the current state-of-the-art in software tools and the organizational structure of the various SDOs and other associations, a number of observations can be made and open issues identified relating to the questions posed at the start of this paper.

A. SOFTWARIZATION VS. PERFORMANCE

As the target speeds of IMT-2020 and future systems per UE increases, it must be assumed that the aggregate throughput

TABLE 3. SDO work area comparison.

	Protocol	Edge	Net Slicing	Architecture	Interoperability
IMT-2020				✓✓✓	✓
FGNET-2030				✓✓✓	
3GPP	✓	✓✓	✓✓	✓✓✓	✓✓✓
ETSI NFV	✓		✓✓✓	✓	✓✓
ETSI MEC		✓✓✓		✓	✓✓
ETSI NGP	✓✓✓				✓✓
ETSI ZSM				✓✓	✓✓✓
IETF DMM	✓✓✓		✓✓	✓	✓
IETF 5gangip	✓✓✓				✓✓✓
IETF netslice			✓✓✓		✓
IETF raw	✓		✓	✓	✓
IRTF coinrg	✓✓	✓	✓	✓✓✓	
IRTF icnrg		✓✓✓		✓✓✓	
IRTF panrg	✓✓	✓✓		✓✓	

✓✓✓ Totally
✓✓ Mostly
✓ Partially

of the software-based systems will be required to increase. Research must therefore continue to investigate how the benefits of moving from for-purpose hardware to software solutions can overcome the current issues in performance. In the same way that pure hardware solutions are seen as inflexible, pure software solutions appear not to be able to deliver the full range of performance required per use-case and therefore consideration of hybrid software/hardware solutions may be appropriate.

Context-awareness can also be used to determine whether all data is required to be only handled by software or only by hardware and research to understand how to identify and then appropriately treat flows is required. Future work into understanding the intersection between hardware and software solutions should focus on how to deliver an appropriate outcome based on user need.

Whilst softwarization alone of the network functions does not negatively impact performance it is the manner in which this softwarization is achieved that has the most impact. For example a software-based UPF in a bare-metal system with highly-tuned parameters is likely to perform as-well as a custom appliance; however, a virtualized version of the same software deployed as a VNF will suffer from performance degradation.

Furthermore, the architecture of the total softwarized environment becomes important in that certain traffic flows may be impacted to a greater degree than others – for example, north-south flows may not be impacted as much as east-west flows in certain combinations of virtualization and network architecture.

Thus the benefit of the flexibility of softwarization coupled with virtualization must be balanced with the additional complexity introduced.

Softwarized solutions around edge deployment offer closer applications and data in order to improve latency – however, new developments in technology such as hollow-fiber (Bradley *et al.* [169]) are also looking to address latency on a global scale.

The authors suggest that not all 5G use-cases will be served by purely softwarized or purely appliance based architectures – instead, hybrid solutions using both software and hardware will be deployed with services split in order to provide the required performance guarantees.

B. SOFTWARIZATION VS. STANDARDIZATION

The proliferation of software tools and resources has led to a number of issues for ease-of-deployment. On the one-hand, the ability to develop quickly has been good in that new software becomes available much more quickly than had been the case with hardware appliances. To that ends, an entire 5G packet core can be constructed using freely-available tools.

On the other hand, this has led to different level of maturity especially in respect of adherence to the standards. Many standards exist to promote inter-operability and this is sometime compromised when using open-source software solutions. There is also the risk of software becoming isolated from the main standardization tracks. This is fine all the time that support is readily available – however, it has been noted that whilst initial enthusiasm is strong for many software solutions, in the open-source world, especially in the FOSS model, support is provided on a best-efforts basis and can be highly variable over time.

It should also be noted that different SDOs follow different directions – IETF is based on taking existing solutions and building standards around those – the ‘‘rough consensus and

running code'' mantra of the IETF [170] – as opposed to 3GPP which uses an top-down architectural model to which all manufacturers must adhere. Tensions exist between these two models.

C. OPEN-SOURCE VS. "CLOSED"-SOURCE SOFTWARIZATION

Within the mobile industry, standards compliance and the ability to resolve software issues against a Service Level Agreement are key – the emergence of industry bodies such as TIP, CORD or O-RAN help alleviate the reliance on point-solutions instead building known-good architectures from the available components. In addition other models of software design have emerged – the paid-for model which allows for either support of previously FOSS against an SLA or provide closed-source software against a set of standards. The authors consider that both of these options provide the service guarantees which large telecoms providers would need to see in order to consider a fully-softwarized network.

D. OPEN ISSUES

1) RESOURCE SHARING

In the pre-SDN monolithic networks, the resources used were fixed, allocated to the service. In the SDN paradigm, resources are shared; this can be with temporary allocation to slices or could be to multiple tenants. In the hybrid hardware/software model, one piece of hardware could be required to be controlled by several virtual instances.

Sharing resources in this manner requires the implementation of complex scheduling systems that understand the desired outcomes per-slice – today, the slice allocation in 5G is static by the NSSI in the 5G Attach procedure and research is required to understand how traffic can be allocated in a more granular fashion to the available resources.

Coupled with this are the requirements are service isolation and security.

As networks evolve more to the cloud (Network-as-a-Service) model, it is envisaged that future core infrastructure will be predominantly run on shared components and the ability to provision multiple slices with agreed SLAs will be important.

2) MOBILITY MANAGEMENT

The 5G attachment procedure anchors the end-user with a particular network and hides the mobility from the IP layer – GTP at layer 2 provides tunnels to a fixed egress point in the network.

In future networks where the core is made-up of several different network instances, it may not be possible or desirable to build such layer 2 tunnels; for instance, there may be no layer 2 connectivity between the networks or the domain of control between the networks be isolated. Additionally, rather than have a single anchor point, the demands of some applications may benefit from multiple anchor points to take advantage of local network peering (for example, to connect to a specific data-center for content as in the Multi-Access Edge Compute, MEC model) or the requirements of the user may be such that the device is fixed in which case no anchor point is required.

The current GTP-based mobility services are a ''one-size fits all'' approach with the drawback that complex forwarding and sub-optimal paths will exist to provide simple mobility within a network domain.

Research needs to study the problem of mobility by use-case and by network architecture, both within a single network and where a service is built from several networks. In cases where fast mobility is required across multiple networks, some kind of predictive method may be required in conjunction either with known or learnt physical mobility patterns.

3) APPLICATION OF NEW NETWORK PROTOCOLS

Whilst IP is likely to remain the dominant protocol for some considerable time, new use-cases and application are being built around non-IP protocols. For example, many IoT applications use very small payloads which are carried in large packets with multiple headers. This may be leading to inefficient architectures or excessive use of power for non-data-carrying elements of the packets.

New techniques are required to enable the transport of these protocols over both existing and future packet-core networks, especially where interfaces and software have been previously designed around the IP stack.

4) AI IN 5G

To-date the application of AI in 5G has concentrated around the modelling and solving of RF resource allocation problems in 5GNR. Work in both ITU-T [171] and 3GPP [172] have identified areas where AI could be useful. However, the resource allocation issue is apparent in the packet network from areas such as predictive handover to pre-positioning of content based on possible UE trajectory. Network performance modelling to ensure an SLA could also potentially be addressed by AI. As the softwarization of the packet core presents many possible permutations of architecture and packet flow it can be classified as an NP-Hard problem and therefore ideal to be solved by AI techniques.

VI. SUMMARY

Softwarization offers the possibility of delivering new services over the 5G by enabling flexibility in programming and deployment of core network components, items previously fixed in for-purpose scenarios.

However, limits to the flexibility occur both in terms of the impact of standardization, designed primarily to create interoperable solutions, and limitation on performance of the underlying components.

Complexity in terms of the number of projects which have been initiated makes building solutions challenging, and a number of architectural-lead projects have created blue-prints

by encompassing several of the open-source projects in specific ways.

With the different operational models of the SDOs, this document has summarized each of these areas, explaining how best to navigate this vast area.

In considering the current state-of-the-art in software tools and the organizational structure of the various SDOs and other associations, a number of observations can be made and open issues identified relating to the questions posed at the start of this paper.

REFERENCES

[1] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2014.

[2] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.

[3] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 462–476, Sep. 2016, doi: 10.1109/TNSM.2016.2597295.

[4] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 60–65, Mar. 2018, doi: 10.1109/MCOMSTD.2018.1700072.

[5] V.-G. Nguyen, A. Brunstrom, K.-J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1567–1602, 3rd Quart., 2017, doi: 10.1109/COMST.2017.2690823.

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[7] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2815638.

[8] U. Habiba and E. Hossain, "Auction mechanisms for virtualization in 5G cellular networks: Basics, trends, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2264–2293, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2811395.

[9] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1409–1434, 2nd Quart., 2019, doi: 10.1109/COMST.2018.2884835.

[10] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges," *Comput. Netw.*, vol. 146, pp. 65–84, Dec. 2018, doi: 10.1016/j.comnet.2018.09.005.

[11] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984, doi: 10.1016/j.comnet.2019.106984.

[12] *System Architecture for the 5G System 15.3.0*, document 23.501, 3GPP, 2018.

[13] L. Conteras and D. Lopez, "A network service provider perspective on network slicing," in *Proc. IEEE Softwarization*, Jan. 2018.

[14] N. Davies and P. Thompson, "Challenges of network slicing," in *Proc. IEEE Softwarization*, Jan. 2017.

[15] *Making 5G a Reality*, NEC Corp., Tokyo, Japan, 2018.

[16] M. Iwamura, "NGMN view on 5G architecture," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–5.

[17] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[18] J. Halpern and C. Pignataro, *Service Function Chaining (SFC) Architecture*, document REC 7665, 2015.

[19] A. Farrel, "Recent developments in service function chaining (SFC) and network slicing in backhaul and metro networks in support of 5G," in *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Bucharest, Romania, Jul. 2018, pp. 1–4.

[20] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The segment routing architecture," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1–6.

[21] A. Cooper. (Jun. 2017). *Internet Engineering Task Force and 5G Standardization*. [Online]. Available: https://www.3gpp.org/ftp/Information/presentations/Presentations_2017/3GPP%20-%20IETF%20and%205G%20v2.pdf

[22] P. Quinn and J. Guichard, "Service function chaining: Creating a service plane via network service headers," *Computer*, vol. 47, no. 11, pp. 38–44, Nov. 2014.

[23] W. Haeffner, J. Napper, M. Stiemerling, D. Lopez, and J. Uttaro, "Service function chaining use cases in mobile networks," IETF, Fremont, CA, USA, Tech. Rep., 2018.

[24] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.

[25] F. Callegati, W. Cerroni, C. Contoli, and G. Santandrea, "Dynamic chaining of virtual network functions in cloud-based edge networks," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, Apr. 2015, pp. 1–5.

[26] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, Nov. 2016.

[27] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, p. 1, 2007.

[28] ITU-T. (1994). *Q.700: Introduction to CCITT Signalling System No 7*. Accessed: Oct. 20, 2018. [Online]. Available: http://www.itu.int/rec/T-REC-Q.700/en

[29] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69, 2008.

[30] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[31] B. Sayadi, M. Gramaglia, V. Friderikos, D. Hugo, P. Arnold, M.-L. Alberi-Morel, M. A. Puente, V. Sciancalepore, I. Digon, and M. R. Crippa, "SDN for 5G mobile networks: NORMA perspective," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw.*, in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2016, pp. 741–753.

[32] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 21–27, Sep. 2017.

[33] H. Selvi, S. Guner, G. Gur, and F. Alagoz, "The controller placement problem in software defined mobile networks (SDMN)," in *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*, M. Liyanage, A. Gurtov, and M. Ylianttila, Eds. Hoboken, NJ, USA: Wiley, 2018, pp. 128–147.

[34] M. Latah and L. Toker, "Application of artificial intelligence to software defined networking: A survey," *Indian J. Sci. Technol.*, vol. 9, no. 44, no. 1–7, Nov. 2016.

[35] M. He, P. Kalmbach, A. Blenk, W. Kellerer, and S. Schmid, "Algorithm-data driven optimization of adaptive communication networks," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–6.

[36] P4.org. (2017). *P4~16~ Language Specification*. [Online]. Available: https://p4.org/p4-spec/docs/P4-16-v1.0.0-spec.html

[37] B. Briscoe, D. Clarke, P. Willis, A. Reid, and P. Veitch, "Network functions virtualisation," Tech. Rep., 2013.

[38] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Ferger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, and H. E. A. Deng, "Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action," Tech. Rep., 2012.

[39] C. G. Kominos, N. Seyvet, and K. Vandikas, "Bare-metal, virtual machines and containers in OpenStack," in *Proc. 20th Conf. Innov. Clouds, Internet Netw. (ICIN)*, Mar. 2017, pp. 36–43.

[40] S. R. U. Kakakhel, L. Mukkala, T. Westerlund, and J. Plosila, "Virtualization at the network edge: A technology perspective," in *Proc. 3rd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Apr. 2018, pp. 87–92.

[41] A. Mayoral, R. Vilalta, R. Casellas, R. Martínez, and R. Munoz, "Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances," in *Proc. 42nd Eur. Conf. Opt. Commun. (ECOC)*, vol. 16, 2018, pp. 509–512.
[42] D. Lake, G. Foster, S. Vural, Y. Rahulan, B.-H. Oh, N. Wang, and R. Tafazolli, "Virtualising and orchestrating a 5G evolved packet core network," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–5.
[43] S. Peterson. (2011). *What's the Difference Between Open Source Software and Free Software?* [Online]. Available: https://opensource.com/article/17/11/open-source-or-free-software
[44] OPNFV. (2020). *Home—OPNFV*. [Online]. Available: https://www.opnfv.org/
[45] L. Zhu, M. M. Karim, K. Sharif, F. Li, X. Du, and M. Guizani, "SDN controllers: Benchmarking & performance evaluation," 2019, *arXiv:1902.04491*. [Online]. Available: https://arxiv.org/abs/1902.04491
[46] OpenDaylight. (2020). *OpenDaylight*. [Online]. Available: https://www.opendaylight.org/
[47] Akka. (2019). *Akka: Build Concurrent, Distributed, and Resilient Message-Driven Applications for Java and Scala*. [Online]. Available: https://akka.io/
[48] ONOS. (2020). *Open Network Operating System*. [Online]. Available: https://onosproject.org/
[49] The Linux Foundation. (2020). *The Linux Foundation—Supporting Open Source Ecosystems*. [Online]. Available: https://www.linuxfoundation.org/
[50] (2020). *Ryu SDN Framework*. [Online]. Available: https://osrg.github.io/ryu/
[51] A. L. Stancu, S. Halunga, A. Vulpe, G. Suciu, O. Fratu, and E. C. Popovici, "A comparison between several software defined networking controllers," in *Proc. 12th Int. Conf. Telecommun. Mod. Satell., Cable Broadcast. Services (TELSIKS)*, Nis, Serbia, 2015, pp. 223–226.
[52] Apache. (2019). *Apache ZooKeeper*. [Online]. Available: https://zookeeper.apache.org/
[53] OpenVSwitch. (2020). *OpenVSwitch*. [Online]. Available: https://www.openvswitch.org/
[54] Microsoft. (2017). *Overview of Single Root I/O Virtualization (SR-IOV)*. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/network/overview-of-single-root-i-o-virtualization–sr-iov-
[55] E. L. Fernandes, E. Rojas, J. Alvarez-Horcajo, Z. L. Kis, D. Sanvito, N. Bonelli, C. Cascone, and C. E. Rothenberg, "The road to BOFUSS: The basic OpenFlow user-space software switch," 2019, *arXiv:1901.06699*. [Online]. Available: http://arxiv.org/abs/1901.06699
[56] G. Bianchi, M. Bonola, A. Capone, and C. Cascone, "OpenState," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 44–51, Apr. 2014.
[57] R. Bifulco and G. Retvari, "A survey on the programmable data plane: Abstractions, architectures, and open problems," in *Proc. HPSR*, Jun. 2018, pp. 1–7.
[58] N. Zilberman, Y. Audzevich, G. A. Covington, and A. W. Moore, "NetFPGA SUME: Toward 100 Gbps as research commodity," *IEEE Micro*, vol. 34, no. 5, pp. 32–41, Sep. 2014.
[59] A. C. Baktir, A. Ozgovde, and C. Ersoy, "Implementing service-centric model with p4: A fully-programmable approach," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–6.
[60] H. Wang, R. Soulé, H. T. Dang, K. S. Lee, V. Shrivastav, N. Foster, and H. Weatherspoon, "P4FPGA: A rapid prototyping framework for p4," in *Proc. Symp. SDN Res.*, Apr. 2017, pp. 122–135.
[61] (2017). *NetFPGA/P4-NetFPGA-Public*. Accessed: Oct. 22, 2018. [Online]. Available: https://github.com/NetFPGA/P4-NetFPGA-public
[62] DPDK. (2020). *Home—DPDK*. [Online]. Available: https://www.dpdk.org/
[63] P. Bosshart, G. Varghese, D. Walker, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, and A. Vahdat, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, 2014.
[64] Open Networking Foundation. (2020). *Software-Defined Networking (SDN) Definition—Open Networking Foundation*. [Online]. Available: https://www.opennetworking.org/sdn-definition/
[65] The P4.org API Working Group. (2019). *P4Runtime Specification*. [Online]. Available: https://p4.org/p4runtime/spec/v1.0.0/P4Runtime-Spec.html
[66] (2020). *gRPC*. [Online]. Available: https://grpc.io/
[67] OpenStack. (2020). *Build the Future of Open Infrastructure*. [Online]. Available: https://www.openstack.org/
[68] OpenStack. (2020). *OpenStack Docs: Introduction: A Bit of OpenStack History*. [Online]. Available: https://docs.openstack.org/project-team-guide/introduction.html
[69] (2020). *Web Archive Nova*. [Online]. Available: https://web.archive.org/web/20100620230941/http://novacc.org
[70] (2020). *OpenStack Docs: Logical Architecture*. [Online]. Available: https://docs.openstack.org/install-guide/get-started-logical-architecture.html
[71] Kubernetes. (2020). *Production-Grade Container Orchestration*. [Online]. Available: https://kubernetes.io/
[72] Docker. (2020). *Empowering App Development for Developers | Docker*. [Online]. Available: https://www.docker.com/
[73] ONOS Project. (2020). *SONA-CNI Installation—ONOS—Wiki*. [Online]. Available: https://wiki.onosproject.org/display/ONOS/SONA-CNI+Installation
[74] Flexera. (2019). *RightScale 2019 State of the Cloud Report From Flexera*. [Online]. Available: https://info.flexera.com/SLO-CM-WP-State-of-the-Cloud-2019
[75] The Apache Software Foundation. (2020). *Apache CloudStack*. [Online]. Available: https://cloudstack.apache.org
[76] N. Vazquez, "Open vSwitch and DPDK support in CloudStack," ApacheCon North Amer., Las Vegas, NV, USA, Tech. Rep., 2019.
[77] OpenNebula Systems. (2020). *OpenNebula*. [Online]. Available: https://opennebula.io/
[78] OpenNebula. (2019). *OpenNebula 5.10 'Boomerang' Beta is Out!* [Online]. Available: https://opennebula.io/opennebula-5-10-beta-boomerang-is-out/
[79] Proxmox Server Solutions. (2020). *Proxmox Virtual Environment*. [Online]. Available: https://www.proxmox.com/en/proxmox-ve
[80] S. Shin. (2018). *SONA for Kuryr-Kubernetes—ONOS—Wiki*. [Online]. Available: https://wiki.onosproject.org/display/ONOS/SONA+for+Kuryr-Kubernetes
[81] K. Flynn. (2019). *Release 15*. [Online]. Available: https://www.3gpp.org/release-15
[82] (2020). *Open5GCore*. [Online]. Available: https://www.open5gcore.org/
[83] Fraunhofer. (2020). *5G Playground*. [Online]. Available: https://www.fokus.fraunhofer.de/go/en/fokus_testbeds/5g_playground
[84] M. Corici, M. Emmelmann, M. Hauswirth, and T. Magedanz, "Paving the way for local and industrial 5G networks and testbeds," *ERCIM News*, vol. April, no. 117, pp. 7–9, 2019.
[85] 5Genesis Project. (2020). *5GENESIS*. [Online]. Available: https://5genesis.eu
[86] (2020). *5G-VINNI*. [Online]. Available: https://www.5g-vinni.eu
[87] (2020). *SATis5 Demonstrator for Satellite-Terrestrial Integration in the 5G Context*. [Online]. Available: http://satis5.eurescom.eu
[88] (2020). *Free5GC*. [Online]. Available: https://www.free5gc.org/
[89] (2020). *NextEPC*. [Online]. Available: https://nextepc.org/
[90] C.-W. Liao, F. J. Lin, and Y. Sato, "Evaluating NFV-enabled network slicing for 5G core," in *Proc. 21st Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Daegu, South Korea, 2020, pp. 401–404.
[91] L. Tomaszewski, S. Kuklinski, and R. Kolakowski, "A new approach to 5G and MEC integration," in *Proc. 16th Int. Conf. Artif. Intell. Appl. Innov.*, 2020, pp. 15–24.
[92] EURECOM. (2020). *OpenAirInterface—5G Software Alliance for Democratising Wireless Innovation*. [Online]. Available: https://www.openairinterface.org/
[93] A. Kapadia and N. Chase, *Understanding OPNFV*. Sunnyvale, CA, USA: Mirantis, 2017.
[94] (2020). *ORAN Alliance*. [Online]. Available: https://www.o-ran.org/
[95] Telecom Infra Project. (2020). *Home—Telecom Infra Project*. [Online]. Available: https://telecominfraproject.com/
[96] I. Morris. (2018). *TIP, ORAN Alliance Poised to Join Forces on Open RAN*. [Online]. Available: https://www.lightreading.com/mobile/fronthaul-c-ran/tip-oran-alliance-poised-to-join-forces-on-open-ran/d/d-id/746815
[97] OpenCORD. (2020). *CORD, Central Office Re-Architected as a Datacenter: The Killer App for SDN & NFV*. [Online]. Available: https://opencord.org/
[98] OpenCORD. (2020). *FAQ—CORD—OpenCORD Wiki*. [Online]. Available: https://wiki.opencord.org/display/CORD/FAQ

[99] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem," in *Proc. 4th Workshop All Things Cellular, Oper., Appl., Challenges (AllThingsCellular)*, 2014, pp. 33–38.

[100] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 98–106.

[101] C. Lim, "Mellanox efficient virtual network for service providers," Mellanox, Singapore, Tech. Rep., 2016.

[102] D. Raumer, S. Bauer, P. Emmerich, and G. Carle, "Performance implications for intra-node placement of network function chains," in *Proc. IEEE 6th Int. Conf. Cloud Netw. (CloudNet)*, Sep. 2017, pp. 1–6.

[103] R. Kawashima, H. Nakayama, T. Hayashi, and H. Matsuo, "Evaluation of forwarding efficiency in NFV-nodes toward predictable service chain performance," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 920–933, Dec. 2017.

[104] N. Pitaev, M. Falkner, A. Leivadeasy, and I. Lambadarisy, "Multi-VNF performance characterization for virtualized network functions," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–5.

[105] C. Ge, D. Lake, N. Wang, Y. Rahulan, and R. Tafazolli, "Context-aware service chaining framework for over-the-top applications in 5G networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 50–55.

[106] S. Lange, A. Nguyen-Ngoc, S. Gebert, T. Zinner, M. Jarschel, A. Kopsel, M. Sune, D. Raumer, S. Gallenmuller, G. Carle, and P. Tran-Gia, "Performance benchmarking of a software-based LTE SGW," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 378–383.

[107] B. Pinczel, D. Gehberger, Z. Turanyi, and B. Formanek, "Towards high performance packet processing for 5G," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2015, pp. 67–73.

[108] S. Hemminger, "East-west virtual machine performance," Tech. Rep., 2018.

[109] J. Mwangama and N. Ventura, "Accelerated virtual switching support of 5G NFV-based mobile networks," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–7.

[110] M. Amaral, J. Polo, D. Carrera, I. Mohomed, M. Unuvar, and M. Steinder, "Performance evaluation of microservices architectures using containers," in *Proc. IEEE 14th Int. Symp. Netw. Comput. Appl.*, Sep. 2015, pp. 27–34.

[111] S. Livi, Q. Jacquemart, D. L. Pacheco, and G. Urvoy-Keller, "Container-based service chaining: A performance perspective," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 176–181.

[112] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges," *Comput. Netw.*, vol. 146, pp. 65–84, Dec. 2018.

[113] *Network Functions Virtualisation (NFV); Acceleration Technologies; Report on Acceleration Technologies & Use Cases*, ETSI, Valbonne, France, 2015.

[114] *Network Functions Virtualisation (NFV);Use Cases*, ETSI, Valbonne, France, 2017.

[115] J. Nam, J. Seo, and S. Shin, "Probius: Automated approach for VNF and service chain analysis in software-defined NFV," in *Proc. Symp. SDN Res. (SOSR)*, Mar. 2018, pp. 1–13.

[116] N. Nikaein, R. Knopp, F. Kaltenberger, L. Gauthier, C. Bonnet, D. Nussbaum, and R. Ghaddab, "Demo: OpenAirInterface: An open LTE network in a PC," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Maui, HI, USA, Sep. 2014, pp. 305–308.

[117] N. Nikaein, R. Knopp, L. Gauthier, E. Schiller, T. Braun, D. Pichon, C. Bonnet, F. Kaltenberger, and D. Nussbaum, "Demo: Closer to cloud-RAN: RAN as a service," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Paris, France, Sep. 2015, pp. 193–195.

[118] P. Veitch, T. Long, and P. Hutchinson. (Jan. 6, 2017). NFV performance optimization for virtualized customer presmises equipment. Intel. [Online]. Available: https://software.intel.com/en-us/articles/nfv-performance-optimization-for-vcpe

[119] B.-H. Oh, S. Vural, Y. Rahulan, N. Wang, and R. Tafazolli, "Performance evaluation of a virtualized 5G core network in indoor environments," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Rome, Italy, Jun. 2018, pp. 1–6.

[120] *RECOMMENDATION ITU-R M.2083-0IMT Vision-Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, ITU, Geneva, Switzerland, 2015.

[121] ITU-R. (2020). *ITU-R Study Group WP5D*. [Online]. Available: https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/Pages/default.aspx

[122] ITU-T. (2017). *SG13: Future Networks, With Focus on IMT-2020, Cloud Computing and Trusted Network Infrastructures*. [Online]. Available: https://www.itu.int/en/ITU-T/studygroups/2017-2020/13/Pages/default.aspx

[123] ITU-T. (2019). *Focus Group Net2030*. [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/net2030/Pages/default.aspx

[124] Cisco. (2016). *VNI Complete Forecast Highlights*. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf

[125] *Technical Specification Group Services and System Aspects; 3G Security; Lawful Interception Architecture and Functions (Release 15)*, document, Sophia Antipolis, 3GPP, 2019.

[126] IETF. (2011). *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*. [Online]. Available: https://tools.ietf.org/html/rfc6071

[127] (2020). *3GPP*. [Online]. Available: https://www.3gpp.org/

[128] (2020). *ARIB*. [Online]. Available: http://www.arib.or.jp/

[129] (2020). *ATIS*. [Online]. Available: http://atis.org/

[130] (2020). *CCSA*. [Online]. Available: http://www.ccsa.org.cn/english/

[131] (2020). *ETSI*. [Online]. Available: http://www.etsi.org

[132] TSDSI. (2020). *Telecom Standards Development Society, India (TSDSI)*. [Online]. Available: https://tsdsi.in/

[133] TTA. (2020). *Telecommunications Technology Association*. [Online]. Available: https://www.tta.or.kr/English/

[134] TTC. (2020). *Telecommunication Technology Committee*. [Online]. Available: https://www.ttc.or.jp/e

[135] K. Flynn. (2018). *Release 17*. [Online]. Available: https://www.3gpp.org/release-17

[136] K. Flynn. (2020). *Release 16*. [Online]. Available: https://www.3gpp.org/release-16

[137] E. Guttman, *System and Core Network Aspects*, document, Sophia Antipolis, 3GPP, 2018.

[138] D. Seo, *Evolution and Standardization of Mobile Communications Technology*. Hershey, PA, USA: IGI Global, 2013.

[139] *LTE;General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access 16.0.0*, document 23.401, 3GPP, 2018.

[140] U. Mulligan. (2020). *ETSI—Standards for NFV—Network Functions Virtualisation | NFV Solutions*. Accessed: Feb. 9, 2020. [Online]. Available: https://www.etsi.org/technologies/nfv

[141] S. Dahmen-Lhuissier. (2020). *ETSI—Multi-Access Edge Computing—Standards for MEC*. [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing

[142] S. Dahmen-Lhuissier. (2020). *ETSI—Next Generation Mobile Networks | Next Generation Protocols*. [Online]. Available: https://www.etsi.org/technologies/next-generation-protocols

[143] S. Dahmen-Lhuissier. (2020). *ETSI—ZSM—Zero Touch Network & Service Management*. [Online]. Available: https://www.etsi.org/technologies/zero-touch-network-service-management

[144] Internet Society. *Home | Internet Society*. [Online]. Available: https://www.internetsociety.org

[145] (2020). *IETF*. [Online]. Available: https://www.ietf.org

[146] L. Masinter. (1998). *Internet Standards for the Web—Tutorial Notes*. [Online]. Available: https://larrymasinter.net/talks/www7std-notes.pdf

[147] ETSI. (2020). *Universal Mobile Telecommunications System (UMTS); LTE; General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)*. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129200_129299/129281/09.03.00_60/

[148] S. Homma, T. Miyasaka, S. Matsushima, and D. Voyer. (2018). *User Plane Protocol and Architectural Analysis on 3GPP 5G System*. [Online]. Available: https://tools.ietf.org/html/draft-hmm-dmm-5g-uplane-analysis-01

[149] IETF. (2020). *Distributed Mobility Management (DMM)*. [Online]. Available: https://datatracker.ietf.org/group/dmm/about/

[150] H. Chan, X. Wei, J. Lee, S. Jeon, and C. Bernardos. (2019). *Distributed Mobility Anchoring*. [Online]. Available: https://www.ietf.org/id/draft-ietf-dmm-distributed-mobility-anchoring-14.txt

[151] S. Matsushima, L. Bertz, M. Liebsch, S. Gundavelli, D. Moses, and C. Perkins. (2018). *Protocol for Forwarding Policy Configuration (FPC) in DMM*. [Online]. Available: https://tools.ietf.org/html/draft-ietf-dmm-fpc-cpdp-12

[152] *Interface Between the Control Plane and the User Plane Nodes*, document TS 29.244 V16.2.0, 3GPP, 2019. [Online]. Available: http://www.3gpp.org/ftp//Specs/archive/29_series/29.244/29244-g20.zip
[153] S. Homma, T. Miyasaka, S. Matsushima, and D. Voyer. (2019). *User Plane Protocol and Architectural Analysis on 3GPP 5G System*. [Online]. Available: https://www.ietf.org/id/draft-ietf-dmm-5g-uplane-analysis-03
[154] 3GPP. (2019). *TSG Core Network and Terminals WG4 (CT WG4)*. [Online]. Available: https://www.3gpp.org/specifications-groups/ct/ct4-map-camel-gtp-bch-ss-trfo-ims-gup-wlan/home
[155] K. Bogineni, A. Akhavain, T. Herbert, D. Farinacci, A. Rodriguez-Natal, G. Carofiglio, J. Auge, L. Muscariello, P. Camarillo, and S. Homma. (2018). *Optimized Mobile User Plane Solutions for 5G*. [Online]. Available: https://tools.ietf.org/html/draft-bogineni-dmm-optimized-mobile-user-plane-01.
[156] S. Matsushima, C. Filsfils, M. Kohno, P. Camarillo, D. Voyer, and C. Perkins. (2019). *Segment Routing IPv6 for Mobile User Plane*. [Online]. Available: https://ietf-dmm-srv6-mobile-upla
[157] D. Farinacci, P. Pillay-Esnault, and U. Chunduri. (2018). *LISP for the Mobile Network*. [Online]. Available: https://tools.ietf.org/html/draft-farinacci-lisp-mobile-network-03
[158] T. Herbert and K. Bogineni. (2018). *Identifier Locator Addressing for Mobile User-Plane*. [Online]. Available: https://tools.ietf.org/html/draft-herbert-ila-mobile-01
[159] J. Auge, G. Carofiglio, L. Muscariello, and M. Papalini. (2019). *Anchorless Mobility Through hICN*. [Online]. Available: https://tools.ietf.org/html/draft-auge-dmm-hicn-mobility-02
[160] D. V. Hugo and B. Sarikaya. (2019). *IP Issues and Associated Gaps in Fifth Generation Wireless Networks*. [Online]. Available: https://tools.ietf.org/html/draft-hspab-5gangip-atticps-00
[161] IETF. (2020). *Network Slicing (Netslicing)*. [Online]. Available: https://datatracker.ietf.org/wg/netslicing/about/.
[162] IETF. (2020). *Reliable and Available Wireless (Raw)*. [Online]. Available: https://datatracker.ietf.org/wg/raw/about/
[163] (2020). *IRTF*. [Online]. Available: https://irtf.org/
[164] IRTF. (2020). *Computing in the Network Research Group (Coinrg)*. [Online]. Available: https://datatracker.ietf.org/rg/coinrg/about/
[165] IRTF. (2020). *Information-Centric Networking (ICNRG)*. [Online]. Available: https://datatracker.ietf.org/rg/icnrg/about/
[166] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.
[167] IRTF. (2020). *Path Aware Networking RG (Panrg)*. [Online]. Available: https://datatracker.ietf.org/rg/panrg/about/.
[168] T. Enghardt and C. Kraehenbuehl, "A vocabulary of path properties," IETF, San Francisco, CA, USA, Tech. Rep., 2019.
[169] T. D. Bradley, G. T. Jasion, J. R. Hayes, Y. Chen, L. Hooper, H. Sakr, M. Alonso, A. Taranta, A. Saljoghei, H. C. Mulvad, M. Fake, I. A. K. Davidson, N. V. Wheeler, E. N. Fokoua, W. Wang, S. R. Sandoghchi, D. J. Richardson, and F. Poletti, "Antiresonant hollow core fibre with 0.65 dB/km attenuation across the C and L telecommunication bands," in *Proc. 45th Eur. Conf. Opt. Commun. (ECOC)*, 2019, pp. 1–4.
[170] D. Clark, "A cloudy crystal ball—Visions of the future," in *Proc. 24th Internet Eng. Task Force*, Cambridge, MA, USA, 1992.
[171] *Liaison Statement Output on the Results of the 1st Meeting of the ITU-T Focus Group on Machine Learning for Future Networks Including 5G*, ITU-T, Geneva, Switzerland, 2018.
[172] document TS 129 520, 3GPP, ETSI, Sophia Anitpolis, 2018.

DAVID LAKE received the B.Sc. (Eng.) degree in electrical and electronic engineering from City University London, London, U.K., in 1986, and the M.Sc. degree in telecommunications from Birmingham City University, U.K., in 2014. He is currently pursuing the Ph.D. degree with the 5G Innovation Centre, University of Surrey, Guildford, U.K.

From 1998 to 2016, he was an Architect with the CTO Office, Cisco Systems, London, and San Jose, CA, USA. Since 2017, he has been an Engineering Technologist with the Office of the CTO, Dell, London. His research interests include high-performance packet network in the mobile environment and new protocols. He is a regular contributor to IETF and ETSI and holds a number of patents in the area.

Mr. Lake is a member of IET and ISM.

NING WANG (Senior Member, IEEE) received the B.Eng. degree (Hons.) from the Changchun University of Science and Technology, China, in 1996, the M.Eng. degree from Nanyang University, Singapore, in 2000, and the Ph.D. degree from the University of Surrey, in 2004.

He is currently a Professor with the 5G Innovation Centre, Institute for Communication Systems, University of Surrey, U.K. He has been leading a research team at Surrey on intelligent and high-performance networking and service delivery. He has been making active contributions to international standardization bodies, including 3GPP, ITU-T, IETF, and ETSI. His main research interests include 5G and beyond networking, network and service management, mobile content delivery, and the IoT applications.

RAHIM TAFAZOLLI (Senior Member, IEEE) is currently a Professor of electronic engineering. He has been a Professor of mobile and satellite communications with the University of Surrey, since April 2000, the Director of the Institute of Communication Systems (ICS), since January 2010, and the Founder and the Director of the 5G Innovation Centre, since 2012, the world's first dedicated center on 5G research and innovation. He has more than 30 years of experience in digital communications research and teaching. He has authored or coauthored more than 500 research publications. He was the Leader of study on grand challenges in the Internet of Things (IoT), U.K., from 2011 to 2012, for Research Council U.K. (RCUK), and the U.K. Technology Strategy Board (TSB). He was an Advisor to the Mayor of London with regard to the London Infrastructure Investment 2050 Plan. He is regularly invited by governments to advice on national 5G research and strategy and to deliver keynote talks and distinguished lectures to international conferences and workshops.

Prof. Tafazolli was appointed as a Fellow of the Wireless World Research Forum (WWRF), in 2011, in recognition of his personal contributions to the wireless world and the Head of one of Europe's leading research groups. He was a recipient of the 28th KIA Laureate Award, in 2015, for his contribution to communications technology. This royal recognition was awarded to mark the Queen's Diamond Jubilee.

LOUIS (SAM) SAMUEL received the M.Eng. degree in communication engineering and the Ph.D. degree in the application of non-linear dynamics to teletraffic modeling from the Queen Mary College, University of London, in 1995 and 1999, respectively.

He joined Cisco in April 2013, where he is currently the Director of the Cisco's Emerging Technologies and Incubation Group. Since joining Cisco, he has a responsibility for 5G technology evolution, which covered a wide range of areas such as: virtualization and orchestration of its mobility products, autonomic management of cellular networks (via the application of SON technologies), policy and small cell technology evolution, and the application of 5G to enterprises and verticals. He has been investigating, among others, the long-term effects of pandemics on technology accessibility.