Lee, GM

Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges

network

MDPI

Review

Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges

Ehsan Ahvar [1,*], Shohreh Ahvar [2], Syed Mohsan Raza [3], Jose Manuel Sanchez Vilchez [4] and Gyu Myoung Lee [5]

[1] Learning, Data and Robotics Laboratory, ESIEA Graduate Engineering School, 75005 Paris, France
[2] ISEP-Institut Supérieur d'Électronique de Paris, 75006 Paris, France; shohreh.ahvar@isep.fr
[3] Department of Computing, Abasyn University, Peshawar 25000, Pakistan; mohsan.raza@abasyn.edu.pk
[4] Orange Labs, 92320 Chatillon, France; jose2.sanchez@orange.com
[5] School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, UK; g.m.lee@ljmu.ac.uk
* Correspondence: ehsan.ahvar@esiea.fr

Abstract: In recent years, the number of objects connected to the internet have significantly increased. Increasing the number of connected devices to the internet is transforming today's Internet of Things (IoT) into massive IoT of the future. It is predicted that, in a few years, a high communication and computation capacity will be required to meet the demands of massive IoT devices and applications requiring data sharing and processing. 5G and beyond mobile networks are expected to fulfill a part of these requirements by providing a data rate of up to terabits per second. It will be a key enabler to support massive IoT and emerging mission critical applications with strict delay constraints. On the other hand, the next generation of software-defined networking (SDN) with emerging cloud-related technologies (e.g., fog and edge computing) can play an important role in supporting and implementing the above-mentioned applications. This paper sets out the potential opportunities and important challenges that must be addressed in considering options for using SDN in hybrid cloud-fog systems to support 5G and beyond-enabled applications.

Keywords: SDN; cloud computing; 5G; 6G; fog computing; challenge; massive IoT

check for updates

Citation: Ahvar, E.; Ahvar, S.; Raza, S.M.; Manuel Sanchez Vilchez, J.; Lee, G.M. Next Generation of SDN in Cloud-Fog for 5G and Beyond-Enabled Applications: Opportunities and Challenges. *Network* **2021**, *1*, 28–49. https://doi.org/10.3390/network1010004

Academic Editor: Carlos J. Bernardos

Received: 30 May 2021
Accepted: 22 June 2021
Published: 23 June 2021

1. Introduction

The number of connected devices to the internet is sharply increasing. It is predicted that the number of Internet of Things (IoT) devices worldwide will increase from around 8.74 billion in 2020 to more than 25.4 billion IoT devices in 2030 [1]. These devices will transform today's IoT into massive IoT of the future. It is also predicted with the massive IoT, we will be faced with a very high density network (i.e., up to around 1 million devices per square kilometer), where a large number of tasks (services) should be performed and a huge amount of data should be stored and analyzed [2].

As a result, future massive IoT will require high communication and computing capacity to meet the various demands of highly connected devices.

A few years ago, traditional cloud computing, including one or a few large data centers (DCs), was considered to be a promising computing model to store and process a huge volume of data and offer many reliable and mission critical services with strict delay constraints to users.

As these centralized DCs are usually far away from end users and cannot provide ultra-low latency and high bandwidth connectivity, they are not suitable for supporting the emerging applications related to future massive IoT [3]. In addition, many real-time requirements of edge devices with mobility, such as location awareness, are not met by traditional cloud computing.

To overcome all of these technological challenges, Cisco introduced fog computing in 2012. With fog computing, Cisco extended the traditional cloud computing technology paradigm to the edge of the network [4,5]. In other words, fog computing extends cloud services to the edge of networks to reduce latency through the geographical distribution of IoT application components, and provides support for device mobility [6].

In order to make fog computing (i.e., fog nodes) more intelligent, analytic, and efficient, software-defined networking (SDN), an important network technology, can be integrated with fog [5].

On the other hand, 5G and beyond (i.e., 6G) has been considered to be another key enabler for massive IoT and related applications [7].

In essence, 5G implies rethinking from scratch, due to the abrupt transition concerning new use cases and challenging demands, compared to previous generations. The one-network-fits-all model based on slicing must answer to these demanding and ambitious objectives. Therefore, 5G is conceived to support very extreme and demanding use cases, such as enhanced mobile broadband (eMBB), ultra reliable low latency communications (URLLC), and massive machine type communications (mMTC), not only demanding a data rate of gigabits per second, but also ubiquitous connectivity. Moving to 6G networks, they take for granted an AI-empowered vision (for example, see [8]) to support emerging applications with a data rate of up to terabits per second, such as augmented reality, virtual reality, autonomous and connected drones and vehicles, internet of robots, haptic communications and intelligent and automated machines.

Figure 1 shows three key infrastructure enablers to support such types of future applications (i.e., future generation cellular technologies, cloud-related technologies, and computer network with softwarization).

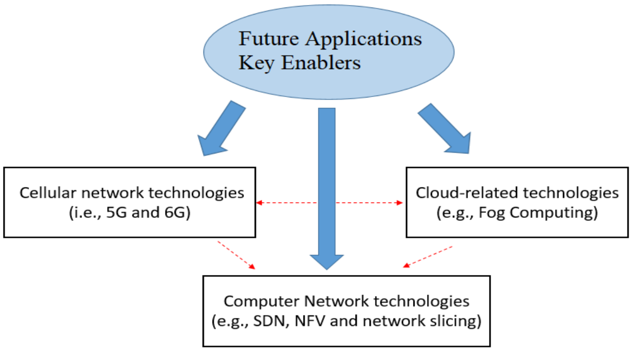
Future Applications
Key Enablers
Cellular network technologies
(i.e., 5G and 6G)
Cloud-related technologies
(e.g., Fog Computing)
Computer Network technologies
(e.g., SDN, NFV and network slicing)

Figure 1. Key enablers for future applications (e.g., massive IoT).

Some enablers, in addition to enabling applications, are considered to be key enablers for other enablers (i.e., reflexive enablers). For example, while cloud computing is considered to be a key enabler for 5G/6G through its underlying technologies, such as Cloud-RAN, cloud computing also needs high-speed 5G/6G networks to offer services with higher quality to its users.

The contributions of this work are summarized as follows:

- This paper introduces some of the key enabling technologies for future applications, such as massive IoT.
- It sets out the potential opportunities of these enabling technologies to support future applications, focusing on SDN for cloud-fog systems to support 5G and beyond-enabled applications.
- It then sets out the potential opportunities and important challenges of these enabling technologies that must be considered and, in particular, challenges related to SDN when integrating it in cloud-fog systems to support 5G and beyond-enabled applications.

Section 2 introduces emerging cloud technologies, focusing on hybrid cloud-fog computing. Section 3 talks about 5G and beyond technologies, focusing on their relation to other key enablers, such as cloud computing and SDN for supporting future applications. We briefly review the concept of SDN in Section 4. Section 5 sets out the current and potential opportunities of SDN for cloud-fog systems to support 5G and beyond-enabled applications. Section 6 highlights the challenges and solutions of using SDN for cloud-fog systems to support 5G and beyond-enabled applications toward the next generation of SDN. Finally, Section 7 concludes our work.

2. Emerging Cloud Technologies

With the origination of (centralized) cloud computing, computation technology has entered into a new era. It can be considered as three services: software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service (IaaS). SaaS provides complete software (e.g., cloud-based email services, social network services, scheduler services) to cloud customers. Application developers can use the platform services provided by cloud PaaS. Finally, IaaS provides virtual machine servers and the related infrastructure to cloud customers. SaaS and PaaS can be implemented upon the usage of IaaS. However, DCs in centralized clouds are geographically centralized and located far from end devices/users. Therefore, they often cannot satisfy requirements of several emerging real-time applications [9–11].

To resolve these issues, besides centralized cloud computing, some new concepts and architectures were proposed. In the following sections, we categorize different cloud-related architectures and terms.

2.1. Cloud-Related Architectures

As already mentioned, in order to support the requirements of different emerging applications, many new cloud-related architectures have been proposed in recent years. These architectures can be generally categorized based on their DC characteristics, such as DC location, number and size.

A cloud-related architecture mainly includes one or more DCs and a number of end users, all connected together through telecommunication networks. In this case, as Figure 2 shows, we can divide cloud-related architectures into four main groups based on DC characteristics: fully centralized (FC), partly distributed (PD), fully distributed with centralized controller (FD-CC), and fully distributed (FD) [12].

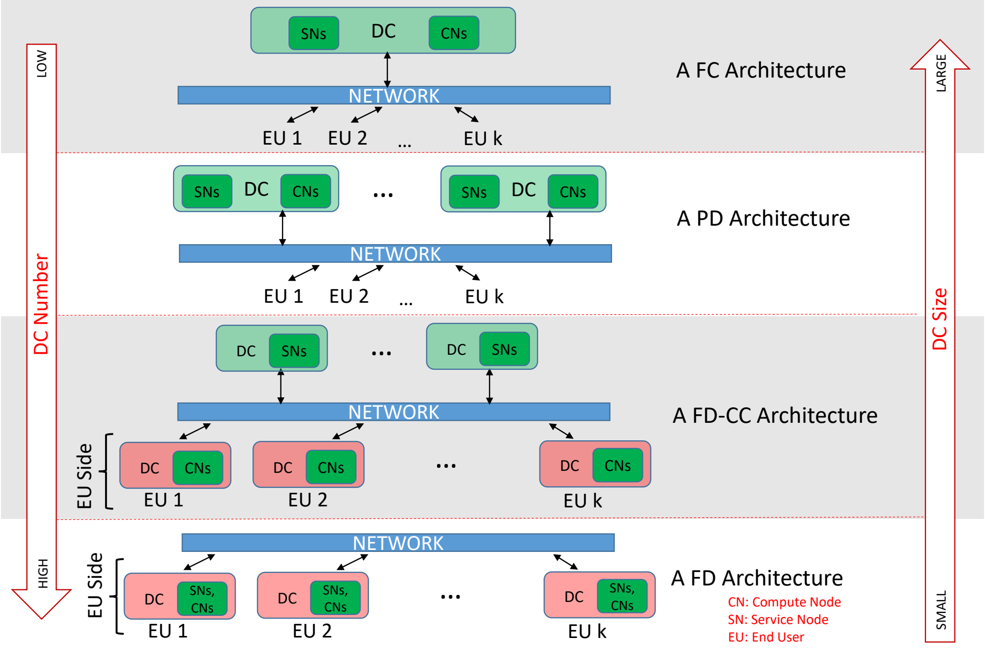

Figure 2. The cloud architecture taxonomy; ranging from fully centralized to fully distributed (inspired by [12]).

In the FC architecture, there is a large-size DC and all end users connect to the central DC and receive services from it. PD architecture includes several geographically distributed DCs, and a telecommunication network connects them together. It also connects end users to DCs. DCs in the PD architecture are usually located closer to end users in comparison to the FC architecture. However, DCs are not still located in end users' premises. The FD-CC architecture includes several DCs that are located on end users premises (i.e., one DC for one or a few end users). However, controller(s) remain in one or more DCs in the core of the network. The DCs that are located in the end users premises are usually in the size of a physical machine (PM). The remaining DCs, in the core of the network, are only used for management aspects and they are not used for computing and storage services. Finally, in the FD architecture, unlike the FD-CC architecture, the management system is also distributed on end users' locations. It means that there is no DC in the core of the network in charge of controlling the resources [12].

2.1.1. Hybrid Cloud-Fog Computing

As already mentioned, the physical distance between the DCs of a cloud service provider and end users (devices) is considered to be a limitation for centralized cloud computing (i.e., the FC architecture). Because of the long distance, centralized cloud computing may not be compatible with the requirements of many latency-sensitive future applications, such as public protection and disaster relief (PPDR) [13], real-time telesurgery [14] and autonomous driving applications.

Fog systems can help cloud computing to provide lower latency by extending the architecture to the edge of the network and allocating some parts of the computations to the edge. We should consider that fog is tightly linked to the existence of a cloud, and it cannot work in a standalone mode [15,16]. While fog resources help in providing the desired quality of service (QoS) for services, resource-limited fog devices cannot be a replacement for the the resource-rich cloud [17]. In fact, fog computing can fill the gap between the cloud and end devices (e.g., IoT nodes) by moving some resources (i.e., computing, storage, networking) and data management to a closer location (i.e., network nodes) to IoT devices. In this case, resource allocation and data management decisions are not only made in the cloud, but can also be made along the IoT-to-cloud path as data traverses to the cloud (preferred to be close to the IoT devices) [18]. It is worth mentioning that bringing the resources closer to the edge can also increase the battery lifetime for edge devices and reduce network traffic in the core of the network [15].

A three-layer general architecture was introduced in several studies (e.g., [17–20]) for fog computing: IoT devices (or edge) layer, fog layer and cloud layer.

The edge layer consists of terminal nodes, embedded systems, sensors and actuators with very limited computation, energy and bandwidth. The fog layer includes intermediate networking devices or nodes (e.g., routers, gateways, switches, and access points) that work on various protocols. These devices are supported with computational and storage capabilities. Finally, cloud DCs that have very rich virtual capabilities in terms of storage and computing are considered to be the last layer [17]. IoT devices (end-users) can communicate with the fog (i.e., fog nodes) through the local area network (LAN). However, the communication between the IoT devices (end-users) and the Cloud is done over the wide area network (WAN), through the fog or otherwise [16].

In these extended hybrid cloud-fog systems, the network scale, topology and configuration are highly heterogeneous, dynamic, and mission-oriented (e.g., temporary hospitals built during the COVID-19 pandemic).

2.1.2. Edge Computing

OpenEdge computing presents Edge computing as computation done at the network edge through small DCs that are located close to users. Edge computing is sometimes incorrectly called fog computing. The OpenFog Consortium makes the distinction that fog computing is hierarchical and offers computing, storage, networking, control, and acceleration at any place from the cloud to devices. However, edge computing tends to be limited to computing at the edge [18,21].

2.1.3. ROOF Computing

The IEEE standard working group working on the project $P1931.1_{TM}$ [22] focuses, among other things, on two main features of semi-autonomous devices: (i) the interactions of such devices with each other and the environment, and (ii) the context-aware decisions of such devices, which are made based on a consensus in a decentralized ecosystem without any human intervention. The result of this attempt was reported as a standard called Real-time Onsite Operations Facilitation (ROOF).

Next-hop connectivity for the things needing real-time context building and decision making as well as efficient backhaul connectivity to remote nodes and networks were addressed in this standard [23]. The utilization of Cloud or Fog properties is minimized with the help of ROOF computing. This goal can be achieved by analyzing the frequently accessed data at the edge of the current user network (e.g., router, access point, server, or related computing platform as suitable) [24].

3. 5G and Beyond

5G allows for several multi-partnership models where traditional services are enriched by adding external third-party functionalities. One example of these business cases is the "operator offer enriched by partner" defined by NGMN [25], where connectivity offered by the operator is enriched through a set of third-party applications. This example manifests the high complexity in such multi-partner services because the responsibility is blurred among several parties. Indeed, each partner must trust each other regarding the decisions made by the rest of the partners to contribute to meet the required overall service quality level.

Moving to 6G, use cases and business models, as pointed out by Dogra et al. in [26], are rather more focused on AI-empowered slice and resource management, haptic communications, internet or robots, tele-robotics and ultra high data density communications, pushing even more forward the technological boundaries. The Finnish 6G Flagship program led by the University of Oulu already published a series of twelve 6G white papers (e.g., [8]) that shed light on main radio opportunities and a vision on emerging use cases, such as holographic teleportation, and unmanned aerial vehicle pervasive connectivity, among others. Table 1 summarizes the requirements for both 5G and 6G.

Table 1. 5G and 6G requirements [27–29] .

Requirement	5G	6G
Energy/bit	NS	1 pJ/bit
Jitter	NS	1 µs
Latency	1 ms	0.1 ms
Traffic Capacity	10 Mbps/m^2	10 Gbps/m^3
Localization Precision	10 cm on 2D	1 cm on 3D
User Experience	50 Mbps 2D	10 Gbps 3D
DL Peal Rate	20 Gbps	1 Tbps
UL Peal Rate	10 Gbps	1 Tbps
Reliability	FER 10^{-5}	FER 10^{-9}
Mobility Support	up to 500 km/h	up to 1000 km/h
Satellite integration	No	Fully
AI	Partial	Fully
Autonomous vehicle	Partial	Fully
Service level	Augmented/Virtual reality	Tactile
Architecture	Massive MIMO	Intelligent surface
Positioning precision	m level	cm level

3.1. Management and Deployment of 5G and Beyond Technology

SDN and network function virtualization (NFV), as complementary technologies for 5G, make the management and deployment of 5G services cost efficient, faster and easier. These are actually the building blocks for more complex services, such as Cloud-RAN, network slicing and intelligence at edge. These services are discussed in this section.

Slicing Architecture through SDN and NFV

5G is expected to support a more flexible and dynamic network using network slicing, which is still a matter of active discussion among the main worldwide standardization organizations and also between telecom operators and vendors. 5G is inherently based on software-defined paradigms and NFV.

Network slicing plays a key role in 5G because the network has to be adapted in real time to at least three different types of usage: machine-to-machine communications, critical communications, mobile broadband, and reliable low latency communications. As such, different types of slices were defined by GSMA [30], such as mobile broadband slice, massive IoT slices, and mission critical IoT slices.

The different 5GPPP projects propose various slicing architectures but remain the same in the essential: a three-layered abstracted model. Indeed, the survey by Fouka et al. in [31] identified a common generic framework for 5G. As it can be seen in Figure 3, a 5G slicing architecture is based on three planes that gather different levels of abstraction: (i) at the top, the service layer, (ii) in the middle, the network function layer, and (iii) at the bottom, the infrastructure layer. The management and orchestration block (MANO) remains transverse to the aforementioned layers. The network function layer embeds all operations considering the life-cycle of the network functions based on the key performance indicators (KPIs) of the corresponding network service. The service layer is directly linked to the business cases and models but is also related to the operational constraints to orchestrate the slice. Each slice is set to comply with a given set of radio access technology (RAT) parameters suitable for that business case. We can imagine that a mobile network operator (MNO) offers a service to a given third party (e.g., a car manufacturer) through a specific end-to-end slice so that it can manage through a dedicated application programming interface (API). The MANO block is composed of the NFV orchestrator (NFVO), the virtual function network manager (VNFM), and the virtual infrastructure manager (VIM). Orchestration is made at three different levels: at the virtual function network (VNF) level through the VNFM, at the network service level through the NFVO, and at the virtual-machine level by the VIM, as defined by the architectural framework by [32]. NFV is inherently hierarchical, with MANO being the transverse brick, able to coordinate the entire orchestration process. The concept of orchestration also comes around when discussing service-oriented architectures, and the management of resources in network infrastructures [33].

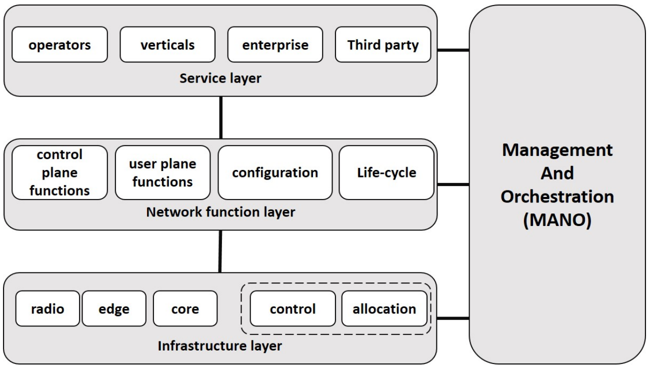

Figure 3. Generic framework across 5G architecture proposals.

This generic architecture gives an idea on how much SDN and NFV are related to 5G. Nevertheless, Ordoñez et al. in [34] went deeper into the 5G architecture implementation with SDN and NFV principles. They joined together the SDN ONF [35] client–server vision with the NFV architectural framework. On one hand, SDN allows to use control plane capabilities to enable slicing, but on the other hand, NFV architectural framework complements SDN by adding the life-cycle capabilities of virtualized network functions. Indeed, they stated that SDN principles are key to implementing 5G slicing due to the recursion property. Abstraction and recursion defined in SDN allow to coordinate successively hierarchical controllers, extending client–server relationships at several levels that are extended with the NFV architectural framework with SDN controllers with tenant and infrastructure roles.

The complexity of 5G network functions obliges to split them into different planes (control plane functions and data plane functions) by the introduction of the software-defined paradigms where abstraction is needed among the different layers. SDN is considered to be an abstraction of network control, where the OpenFlow implementation acts only on the routing capabilities. However, orchestration covers a larger area of network operations. It supports not only network control, but also implies task coordination across multiple entities.

The concept of orchestration also comes around when discussing service-oriented architectures, and management of resources in network infrastructures [33].

Slicing for 6G: Despite flexible allocation of communication resources by slicing in 5G, it can hardly respond to the high expectations of 6G services in terms of delay, reliability and efficiency. Therefore, 6G network slicing technology should be upgraded to support fine-grained, more flexible and intelligent resource allocation. Supporting the cooperation of multiple distributed edge core networks, guaranteeing service migration and the quick deployment and releasing of network slices should be also addressed in the next generation of network slicing [36].

3.2. Cloud-Based 5G/6G Systems

This section describes how cloud-related technologies so far can assist mobile network systems. We briefly review various cloud-based mobile network systems, especially in the domain of RAN architectures, including cloud radio access network (C-RAN), virtualized Cloud-RAN (V-CRAN), and Fog-RAN (F-RAN).

3.2.1. Multi-Access Edge Computing

Multi-access edge computing (MEC), previously called mobile edge computing, is considered to be an extension of mobile computing through edge computing. MEC is defined by European Telecommunications Standards Institute (ETSI) as a platform that provides IT and cloud computing capabilities within RAN in 4G, 5G and beyond, in close proximity to mobile subscribers. It extends edge computing by providing resources close to low-resource mobile devices [18].

3.2.2. Fog RAN

Fog computing can be integrated into the mobile network in the form of RAN, named Fog-RAN (F-RAN). In F-RANs, fog computing resources may be used for caching at the network edge. It can provide faster retrieval of content and lower burden on the front-haul. It is possible to implement F-RAN through 5G related mobile technologies [18].

3.2.3. Cloud-RAN

The key idea behind Cloud-RAN (C-RAN) is to break down the base station functionalities by decoupling the base station into a remote radio head (RRH) and base band unit (BBU). It then centralizes the BBUs from multiple sites into a single geographical location, such as a cloud DC utilizing cloud computing [37]. Enhanced mobility management is also considered to be one of the key advantages of C-RAN [38].

The China Mobile Research Institute (CMRI) announced that by adopting C-RAN, a 15%-reduction in capital expenditure (CAPEX) and a 50%-reduction in operational expenditure (OPEX) can be achieved [38,39].

3.2.4. Virtualized Cloud-RAN (V-CRAN)

In order to better answer requirements of emerging services (e.g., with lower latency), NFV and SDN in the C-RAN can be deployed to virtualize necessary functions and resources (i.e., decoupling data and control planes) [38]. This method of virtualization toward access networks forms a new type of C-RAN, called virtualized-CRAN or V-CRAN.

4. Software Defined Networking (SDN)

In general, traditional networks are complex, and managing them is not an easy task because the control and data planes are combined in network elements (nodes). The paths are determined by the control plane and sent to the data plane. In this traditional method, when the flow management (forwarding policy) has been defined, the only way to modify the policy is via changes to the configuration of the network element. Thus, their operators must configure each individual network element (i.e., switches and routers) separately to express the desired high-level network policies. The configurations are usually done based on several low-level and vendor-specific commands [40–42].

In addition to configuration issues of the network elements, network environments should adapt to traffic changes and endure the dynamics of faults. However, to enforce the required policies, traditional IP networks have poor automatic reconfiguration and response mechanisms [40].

In order to solve the limitations of traditional network infrastructures, SDN was proposed. Unlike traditional IP networks in which the control and data planes are tightly coupled and embedded in the same networking elements, SDN separates the control plane from the data plane. In SDN, control is migrated out of the network elements and into the separate, centralized controller. This control and data planes separation in SDN brings some advantages: (1) it can break the vertical integration, and (2) it can simplify policy enforcement and network (re)configuration and evolution. It is because that the network elements (i.e., switches and routers) become simple forwarding devices, and there is a logically centralized controller [40–42].

We can investigate SDN from various perspectives. Figure 4 presents a tri-fold perspective of SDNs: planes, layers and system design perspectives.

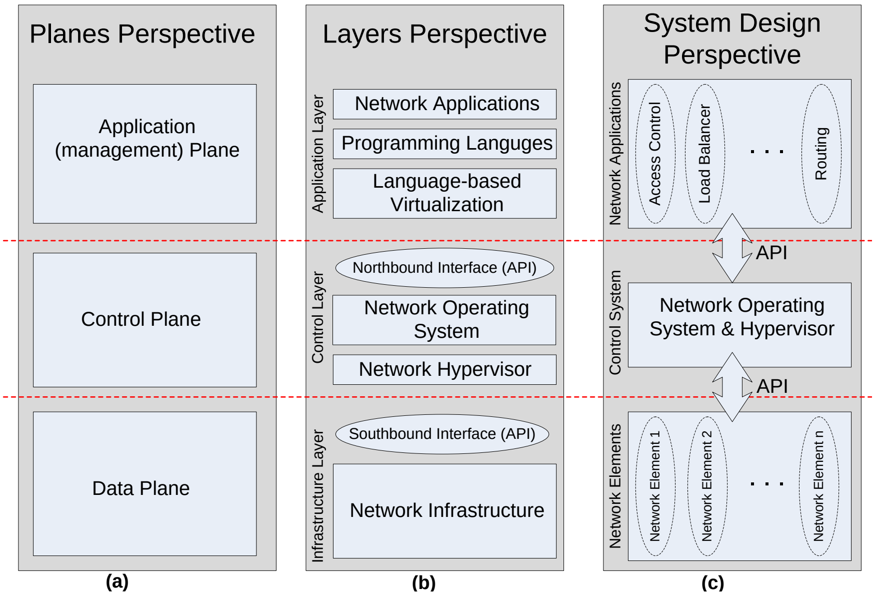

Figure 4. SDN perspectives in planes, layers and system design architecture (inspired by [35,40,41]) (**a**) planes perspective, (**b**) layers perspective, (**c**) system design perspective.

4.1. Planes Perspective

Every computer network is divided into three planes of functionality: management, control and data planes. In a similar way, we can define three planes for SDN as follows [43]:

- The application plane comprises several user applications that talk to the controller to achieve abstraction for a logically centralized controller for making coordinated decisions.
- The control plane, as the SDN brain, manages the whole decision-making process of the network. It consists of all the arrangements to make an intercontroller, data plane to the controller, and application plane to the controller communication.
- The data plane, as the lowest plane in the SDN architecture, directly deals with the physical network infrastructure (switches, routers, and access points).

4.2. Layers Perspective

The Open Networking Foundation (ONF) [35] presented a reference layering model for SDN. The model consists of three layers: infrastructure layer, control layer, and application layer. As Figure 4b shows, in order to have a better description, the layers are divided into the eight following sub-layers [40].

- Network infrastructure sub-layer.
- Southbound interface (or southbound API) sub-layer.
- Network hypervisors sub-layer.
- Network operating system (NOS) sub-layer.
- Northbound interface sub-layer.
- Language-based virtualization sub-layer.
- Programming languages sub-layer.
- Network applications sub-layer.

Notice that while some sub-layers are always present in an SDN deployment (e.g., southbound API, NOS, northbound API and network applications), other sub-layers (e.g., language-based virtualization and hypervisor) are only considered in particular deployments.

4.3. System Design Perspective

As Figure 4c shows, network applications (e.g., access control) for operation and management of the network are represented at the top. Here, a network application refers to a service provided by the network operator. The top tier (i.e., network applications) links with the central tier via an API referred to as the northbound API. Network intelligence is (logically) centralized in the central tier. It consists of the controllers that facilitate setting up and tearing down flows and paths in the network. The central tier links with the bottom tier via an API called the southbound API. The bottom tier comprises the physical network elements, such as switches and routers [41].

4.4. SDN Controllers Design

SDN controllers can be implemented in different deployment models, such as in a centralized or distributed manner, to address different performance requirements. A centralized SDN controller, unlike a distributed one in which each device has a limited knowledge of the network, has comprehensive information of the network topology, traffic flows and the switches load. In other words, it can monitor and manage the entire network. The centralized SDN controller remains connected either in-band or out-of-band to the forwarding elements and defines optimal paths for traffic flows in the network.

In a distributed deployment model, the controllers can be placed vertically, horizontally or as a hybrid (i.e., as peers vertically and horizontally). The design gets more complex when the controllers support both IP and SDN switches. These different designs provide various levels of scalability, consistency, reliability and security.

5. Roles of SDN for Cloud-Fog Enabled 5G and Beyond

The complex, dense and heterogeneous environment nature and constrained resources in cloud-fog systems ask for the automation of management functions. Orchestration functions are required in order to automate the management in such an environment. Considering the special features of cloud-fog, other existing orchestrations, such as the one from cloud, cannot be applied. The service orchestrator must be capable of maintaining resilience, trustworthiness, low latency and acceptable levels of QoS in a dynamic environment of cloud-fog.

The combination of SDN and NFV with fog can facilitate the process of management. In this case, there are three main players: the NFV manager and orchestrator (NFV MANO), the SDN controller and the fog controller. NFV MANO can be used to trigger the necessary flows in the SDN controller, request the needed resources and put the requested services on top of them. It can be also in charge of deploying independent services, using the fog controller [44].

Considering the point that computing takes place on any layer of cloud-fog, depending on the requirements and the the availability of the resources, resource management, as an important part of orchestration, is more challenging in cloud-fog systems. SDN can be used in different resource management aspects as application placement, resource scheduling, task offloading, load balancing, resource allocation and resource provisioning [45]. SDN intervention can be performed in different layers of the IoT-based hybrid cloud-fog system. There exist already research studies for each layer. In this paper, we introduce some works used SDN in hybrid cloud-fog systems, mostly for connecting the cloud layer to the fog layer, or connecting fog nodes together for different use cases.

5.1. SDN in Task Offloading for Cloud-Fog

One of the fits for SDN is in connecting different sections of cloud-fog, especially in efficiently coping with computational offloading. Task offloading refers to getting assistance from other devices in a way that another device performs tasks on behalf of the local device. Task offloading can be in the form of device-to-device offloading [46], device-to-fog offloading [47], fog-to-fog offloading [48] or fog-to-cloud offloading [2].

In the research paper [48], the authors proposed a dynamic offloading mechanism among the fog nodes to address the challenge of static offloading and oblivious link selection in vehicular networks. The incitation for fog-to-fog offloading is raising the computational overhead for a single fog node. Traditionally, the selection of an offloading node based on the most available resources can fabricate various challenges (e.g., congestion, resources under-utilization). In the proposed method, the SDN controller collects the statistics, computational capabilities status, and supported services of a fog node, using the fog agent. When an offloading request is received, the service orchestrator at the controller invokes the appropriate service for the optimal offloading node and a QoS aware path installation. QoS aware path installation is materialized, using the minimum and maximum flow data rate constraints among overloaded and offloading nodes. Thus, the controller supports an overloaded fog node without turning around to static or all offloading fog nodes uncontrollably.

Although related works on computation offloading mostly focus on latency and energy consumption, the topic of reliable computation offloading is commonly left in the dark. One of the very few studies in this topic is the work of Hou et al. [49] for reliable computation offloading in Internet of Vehicle (IoV). The authors also considered the partial offloading, reliability-oriented task allocation and reprocessing mechanism. In the proposed solution, for better resource orchestration, the SDN controller controls the mobile vehicles as mobile edge nodes. In addition, the cluster of edge nodes and SDN are connected by the help of the access Roadside Units (RSUs) through broadband connections. The service request from vehicles or other edge nodes are sent to the access RSU, where the decisions of the SDN controller are executed. Different schemes were considered to report the result where the amount and target of offloading vary. The latency and reliability performance of these schemes were reported. To analyze the performance of proposed solutions, the authors considered three environments, where the links and nodes reliability could be good, poor and mixed. The simulation results showed the high reliability for latency-sensitive applications in an edge computing-enabled software-defined IoV (EC-SDIoV) network.

Focusing on software defined vehicle networks, Lian et al. [50] proposed a framework which deployed the SDN controller in the control layer of the remote cloud. This makes the remote cloud intelligent for the computation of tasks offloaded by a vehicle toward the RSU and addresses the problem of speedy vehicle applications randomly interacting with all the RSU in the region for computational requests. Straightforwardly, the interface among the vehicle communication unit and fog nodes is not aware of any optimal path in advance. Therefore, it leads to costly computation, traffic over oblivious links and delayed feedback from the RSU integrated fog node. Vehicle and fog nodes' transmitted features are cached in the remote cloud DC and are accessible for the controller at any time for routing decisions. RSU generates a request for the optimal path installation from the SDN controller. The controller utilizes the high-performance fog nodes for the real offloaded task by classifying and streamlining the information's table. Consequently, the number of vehicle application's computational requests, processed application content switching (leaving and joining the fog nodes abruptly), and cloud turn around decrease positively. This tends to the optimal computation of IoV jobs by fog nodes.

5.2. *SDN for Resource Allocation in Cloud-Fog*

Ahmad et al. [5] introduced an architecture to address the benefits of energy-saving by combining SDN and the fog environment into one system. To this end, they utilized a dynamic programming scheme in order to optimize the selection of fog nodes. In their proposed architectures, SDN is located between the cloud and fog layers. Using iFogSim [6], the authors showed the ability of the proposed method in minimizing the energy consumed by the fog system while maintaining the service level agreement (SLA) parameters.

Bin et al. [51] proposed a resource allocation mechanism for IoVs in 5G infrastructure. Their mechanism used the cloud-fog system managed by SDN. There are four indicators (i.e., service stability, service delay, energy consumption, load balancing) considered in this paper to deal with the efficient resource allocation for an IoV's demand. The proposed algorithm employed the hierarchical clustering of nodes. Further, in the proposed 5G IoV model, an RSU controller in the fog cluster (managed by the SDN controller) can request the resources to the cloud, based on SDN controller policies. RSU collects and disseminates the connected vehicle and road information to the connected roadside unit controller (RSUC). Due to the uncertain number of job submissions at RSU, there are a number of connected RSUs in the fog cluster. On the other hand, the RSU controller can also request and acquire resources from the adjacent fog cluster to balance the load. The RSU controller aggregates the processed task of a job from a fog node in the cluster and from the cloud node. It is worth mentioning that the historical failure information of a computing entity is maintained in advance to ensure computing node stability during task execution.

Storck et al. [52] introduced a 5G vehicle-to-everything (V2X) ecosystem to support IoV. Their presented ecosystem is based on the SDN concept. The authors evaluated scenarios with the eMBB use case and V2V communications. The results show that the proposal could improve the IoV communications requirements under high-mobility (120 km/h vehicles in a rural environment) and high-density situations (urban scenarios).

Nahida et al. [53] proposed an SDN enabled framework for resource allocation to mobile applications. The authors addressed the delay challenge in computational offloading of mobile devices and efficient resource allocation at the edge of the network along with power constraint considerations. In this framework, firstly, to reduce the central controller burden, edge nodes are used. Secondly, to allocate the efficient computational resources, the reinforcement learning model (Q-learning and enhanced Q-learning algorithms) is employed, which is suitable for dealing with a dynamic demand and frequent resource allocation. Machine learning (ML) aware algorithms work to decrease the latency of resource assignment in massive increasing mobile applications at the edge of the network.

Ahammad et al. [54] proposed an architecture through the combination of fog computing and SDN in order to enhance the QoS in an IoT system. The authors tried to demonstrate how SDN and fog computing can be united effectively in order to recoup for each other's shortcomings and how they can be combined to improve QoS for IoT systems. They also proposed an algorithm (which is dependent on partitioning the SDN virtually) to select the optimal access point and optimal place to process the data. The main objective of this algorithm is to provide improved QoS by partitioning the corresponding fog devices through the SDN controller. This work considers three level of priority for processing data (jobs): time-sensitive (needs real-time processing), less time-sensitive and time-insensitive. In the proposed architecture, there are four following layers: things, fog, SDN and cloud. The things layer includes IoT devices/end user devices. The fog layer is generally divided into two parts: fog node (FN) and aggregate fog node (AFN). Data that are time-sensitive and require real-time processing are executed in the FN sector. Data that are less time-sensitive are processed by AFN and that are time-insensitive are processed by cloud. In this architecture, SDN is utilized as a backbone network for the things–fog–cloud architecture. It is utilized to orchestrate and operate the overall functionalities of the proposed architecture. The FNs and AFNs keep the SDN controller constantly aware of capacities and the location of the things devices to be served. The SDN controller helps with the routing and allocation of resources on the grounds of QoS parameters. The cloud layer is the highest layer of this proposed architecture. It provides a powerful computing and storage capabilities. Ref. [54] was evaluated using iFogsim.

In [55], Ateya et al. provided a framework for integrating heterogeneous IoT networks with 5G networks. The suggested framework deploys three main communication paradigms: MEC, device-to-device communications (D2D), and SDN. The proposed SDN-based architecture includes two main partitions where the heterogeneous IoT networks, D2D-based networks, and RAN are located in the first partition and the second partition consists of the core network (CN). Cloud computing with powerful computing capabilities is available in the CN partition and connected to the control plane of SDN networks. The authors utilized a controller placement algorithm (already proposed in their previous work) to dynamically define the optimum number of SDN controllers required for the networks, according to the current situation of network traffic.

The idea was simulated in the CloudSimSDN simulator [56]. The authors claimed that the proposed framework can reduce the percentage of blocked tasks by an average of 30%, compared to other, traditional IoT networks. In addition, it can reduce the overall energy consumption by an average of 20%, compared to existing IoT networks.

5.3. *Resource Scheduling in Cloud-Fog*

The authors in [57] proposed an architecture for task scheduling in SDN-based IoT-fog networks, in which the fog devices are clustered into virtual organizations (with each of them representing a fog region), and they are interconnected by SDN switches. The central controller of their scheme is located on the cloud gateway. In this work, the authors considered the fog gateway and the cloud gateway as the SDN switch and the SDN controller, respectively.

The study work in [14] proposed an SDN-based cloud-fog architecture for telesurgery in a tactile internet system. Since the performance of robotic telesurgery largely depends on the network performance in terms of latency, jitter and packet loss, the authors used optical network connecting cloud, SDN and fog layers. In the fog layer, a special routing and forwarding method is used for SDN packets. They proposed a scheduling algorithm to meet real-time requirements in the SDN because the traditional algorithms for SDN are not appropriate, effective, or applicable.

SAFE, as the next work in this category, is a secure healthcare ecosystems introduced in [58]. At the edge of the network, the healthcare units exits where thousands of the biosensors are generating the continuous data streams. The primary purpose of medical data computation is improving decisions for health and fast patient recovery. Due to the physical distance of cloud resources from these real-time applications (biosensors, various other hospital applications), SAFE works as a cloud and edge interplay. It supports the computational demand of ever-increasing medical applications for quality of experience (QoE) and QoS constraints. SAFE leverages the lattice-based cryptosystem and deploys global and virtualized SDN controllers to ensure resource allocation, low latency, and services management for increasing healthcare applications in a distributed/multi-region infrastructure.

In [59], the problem of fog node trust is exploited for the security of healthcare unit and patient records. When the cloud-fog is used for smart healthcare, health data need to be securely communicated in a parallel manner. Health data can include various unauthorized intentions, which drives the fog node for biasing behavior and violating the data security policy (e.g., tempering attacks at Fog node, fabricated heath records). Data in the healthcare unit are processed mostly at nearby fog nodes, whereas the data heterogeneity and some real-time or delay-intensive healthcare demands can bottleneck the neighboring fog node. The SDN controller deployed in a distributed manner can manage the load by provisioning the resource effectively and ensure the network security edge efficiently. Authors in [59] also presented a case study that proposes secure privacy aware health data computation with the assistance of fog and cloud nodes.

Junxia et al. [60] also investigated the SDN-enabled edge network for resources and secure services provisioning in a healthcare system. In this paper, a framework is proposed for IoT device authentication that employs the probabilistic k-nearest neighbor (p-KNN). Using p-KNN, an edge server investigates the legitimacy of healthcare IoT devices and SDN performs efficient collaboration among the edge servers that are close computation resources.

The research work [61] addressed the problem of services quality for demands by local residential communities at the edge. Required services often include multimedia services, IoT and video gaming applications in distributed edge access networks. The proposed mechanism introduces function chains to support the differentiated treatment for required services. It also includes a control management and a virtualized management network domain. The control plane in this methodology provides the initiation of NFV, service function chains (SFC) and routing for the application of the residential gateway. The SDN control plane also supports mapping different VNFs to the virtualized management network domain, using APIs in the edge cloud. The residential network management application is responsible for clustering the traffic at a residential gateway interface and steering it to an SFC for responding residential user preferences. Additionally, this application is also in-charge of SDN API, which is responsible for tuning the characteristics of software-defined access network forwarding devices.

5.4. Load Balancing in Cloud-Fog

ScalCon provides a scalable hybrid SDN architecture that distributes the control plane over different controller types (i.e., Fog, Edge, Cloud). ScalCon distributes the load of path computation among all controllers and uses a multi-priority queening model and node parallelism to cut down the synchronization overhead. The simulation results show that ScalCon outperforms existing schemes in terms of path computation time, path setup latency, end-to-end delay and communication overhead [62].

5.5. Resource Provisioning in Cloud-Fog

SDN can support reliable communication management for IoT delay-intensive applications. The application layer in IoT introduces continuous data packets, while the link disruption, due to various reasons, widely occurs when a resource-rich cloud serves these applications. It incurs the QoS and QoE challenges for diverse IoT edge applications. On the other hand, when IoT applications are served by near computational resources (fog nodes), any sort of failure or generic service unavailability requires abrupt flow detouring to meet the expected latency. To enable intelligent, reliable communication, in [63,64], the authors employed an ML algorithm (i.e., k-nearest neighbor regression) in the SDN control function layer for predicting the failure behavior, based on the failure event history. Collecting the network statistical, failure, and resources utilization data is a grace of the SDN controller. So, the objective to confine the ML-aware applications at the controller is to serve the IoT applications over the most reliable communications paths (reliable links and intermediate forwarding devices), ensuring end-to-end latency, failure resiliency and providing better QoS.

Tamil et al. [65] proposed the traffic prediction based resource allocation in 5G infrastructure for IoT applications. The proposed method uses long short term memory (LSTM) at the SDN controller, employed in remote Cloud. To deal with 5G major application use cases (uRLLC), a prediction model supports resource provisioning at the edge of the network. If a local or edge controller is overloaded, based on the predicted load, the cloud controller configures the supported forwarding devices to meet the QoS.

5.6. Application Placement in Cloud-Fog

In NFV-based hybrid cloud-fog systems, application components can be implemented as VNFs and the application itself can be seen as a VNF forwarding graph (FG). As one of the rare works that considered the mobility of fog nodes, ref. [16] proposed an algorithm for application components over the cloud-fog system. The authors used the random waypoint model (RWP) to model the mobility of a fog node. This work tried to minimize the makespan and cost to embed VNF-FGs. Makespan here is the time it takes for the first component to start the execution until the execution of the last component is completed. Although in this work, the role of SDN is not clearly explained, the proposed algorithm can be implemented in the control plane of an SDN-based NFV framework. Another work in [66] studied the same problem in a more heterogeneous environment, including software and hardware NFV platforms, such as virtual machine (VM), docker container and SmartNIC.

Research work [67] contributed to handling dynamic IoV by enabling the hierarchical distributed controllers, where a root controller at a central fog DC manages the underlying controllers and networks. The rest of the controllers are domain and edge controllers, whereas RSUs are directly connected with edge controllers. Edge controllers are deployed at the edge layer, where servers collaborate for fog functions horizontally. However, domain controllers share the network information to the root or global controller in the vertical distributed control architecture. Majorly, the contribution of this work is efficient network management distribution over the root, domain and edge controllers to ensure the least delay for speedy vehicle support. Moreover, the authors also suggested the control channel placement in the fog layers based on the betweenness centrality and modified Louvain algorithm.

6. Challenges and Solutions for Next Generation of SDN

SDN proposes the separation of the control plane from the data plane and the logical centralization of the intelligence in the control plane. However, this makes the control plane the most vulnerable plane, as faults in this plane have worse consequences than faults at the data plane because the SDN controller is embedding the forwarding intelligence. Altekar and Stoica in [68] demonstrated that, under certain conditions, the control plane can be responsible up to for 99% of all bugs in data center applications. An SDN architecture is still subject to some reluctance by different stakeholders, due to a number of factors, listed hereafter along with some solutions.

- Protocol and standardization: The current version of OpenFlow switch specification is 1.5.1, which is openly available for SDN-supported switches configuration. However, SDN standardization in the form of OpenFlow is not scalable, reliable or efficient enough to fully handle all possible scenarios/use cases and managerial operations in hybrid cloud-fog systems, especially in 6G.Enhancing OpenFlow, northbound interface standardization, debugging capabilities of SDN, and east-west interface are some domains which need special consideration in standards [15]. Regarding the east/west standardization improvement, Basem et al. [69] proposed a distributed SDN controller framework (DSF), which is based on real-time publish/subscribe (RTPS) standardization to address the challenge of synchronization in the distributed and heterogeneous control plane. The framework can support the flat, hierarchical, and hybrid/T-model. In the horizontal model, controllers communicate in a peer order; in the hierarchical model, controllers follow a chain of command in a tree structure; and in a hybrid model (comprised of vertical domain control and horizontal global view communication), information travels from domain controllers to peer controllers. The DSF presents the reliable and constant interfacing behavior for holistic topology discovery when the number of controllers is increased. Moreover, DSF supports the homogeneous and heterogeneous control plan entities.

- SDN integration inside NFV ecosystem: The NFV term emerged in 2012 from a meeting of the founding group in Paris to distinguish the topic from SDN. The working group was hosted by ETSI. In the ETSI NFV drafts [70], it is still not clear the role of the SDN controller and its interface with the rest of the NFV ecosystem, especially for distributed environments, such as cloud-fog systems.
- Budget constraints SDN evaluation: Full migration to SDN-based infrastructure can have various challenges, especially budget constraints. Ultimately, in the case of the cloud-fog system, it requires massive SDN component installation and softwarization supported equipment. It can harden the pure SDN deployment. Thus, these systems need a hybrid infrastructure, where the legacy and SDN-based devices, resources, routing protocols, services and virtual environments are interoperable and agreed to support the user expectation. From this agreed paradigm amalgamation emerges hybrid SDN for fog computing, where optimized resource provisioning, updating network information, and latency-aware routing are challenges. One of the hardening effects could be that a global controller (i.e., root controller) is unable to investigate the network topology or switch roles inline with speed as for pure SDN deployed components [63,71].
- Hybrid SDN-traditional IP network architecture challenges: Hybrid SDN-traditional IP network architecture is a solution for now. Investments in hardware are expensive to fully adjust the network to SDN. Therefore, providers are willing to start with the hybrid SDN-traditional IP network architecture. Then, collecting the accurate network topology and commanding the devices that are not directly connected to the controller is a challenge. Furthermore, the separate configuration of both types of devices and accessing the configuration correctness and statistical decision of the controller placement near devices are also next challenges. New SDN solutions supporting multi-layer, multi-vendor operations, ideally with a level of legacy support through open APIs, should be designed to help network providers start adjusting their network to the SDN architecture. Moreover, SDN solutions ideally need to be developed using a cloud-native software base to leverage elastically scalable cloud resources (CPU, memory, and connectivity).
- Government policy adaption: Governments will likely introduce in the future further restrictions for network providers in terms of carbon emission and privacy protection, which also need to be addressed. As an example of IoV, especially in the fog-based 6G networks, is that a vehicle may handover through several small cells that may be untrusted or even compromised, and it is a serious challenge. Here, one of the solutions is preventing linking the vehicle identity with the information of its owner [72].
- Meeting slicing requirement for 6G: As regards 6G, slicing can be considered to be one of the important enabling technologies. As discussed in this paper, network slicing needs to be upgraded for 6G. NFV and SDN as enabler technologies for network slicing should be upgraded in parallel to network slicing upgrade. The controller of NFV and SDN should be upgraded to leverage cognitive services. This enables the scheduling of the network and network functions being integrated into a cognitive

service architecture. Moreover, the virtualization of NFV and SDN should be realized with finer granularity to guarantee flexible resource scheduling.

- Resilience and fault tolerance of the control plane: The most vulnerable aspect of SDN architectures is the control plane, in particular the SDN controller, as the network is at the behest of the SDN controller, which must be protected. The research efforts on resiliency support for SDN (e.g., Da Silva et al. in [73]) categorized them into different planes of the SDN architecture. The most classical solution to provide fault tolerance is redundancy. In SDN, the authors foresaw three potential usages of redundancy: (i) protecting the SDN controller from failures, (ii) protecting the forwarding devices and communication links from link disruption, and lastly (iii) protecting the SDN applications from misconfigurations. There are replication mechanisms (Fonseca et al. in [74]) to protect the control plane from faults and ensure the availability of the SDN controller. Sharma et al. in [75] proposed a fast failure recovery technique for centralized SDN infrastructures, based on a traffic restoration scheme that allows the SDN controller to circumvent faults on certain links at the data plane by proactively sending a set of protection paths. Other works focus on the controller placement, which is tightly linked to ensure SDN controller availability. For instance, Heller et al. in [76] demonstrated that the latency is reduced when the number of controllers is augmented. Muller et al. in [77] combined the placement of the controller with path diversity and recovery mechanisms to ensure the fault tolerance properties in SDN. Ros et al. in [78] studied the controller placement problem from a reliability perspective. The authors provided metrics based on the probability that at least an operational path is available and solved the controller placement problem by imposing as the constraint these path availability metrics.
- Geographically scalability and reliability: The fact that the control plane is logically centralized and often based on the SDN single controller often can cause a single point control channel failure. Nevertheless, the scalability of SDN can be improved by using multi-controller architectures. Multiple SDN controllers can be connected in a flat or hierarchical manner via west–east-bound APIs [9]. However, the controller placement problem in any service oriented architecture is an NP hard problem [67,79]. In addition, still the optimal arrangement, load balancing and demand nature conformed multi controller placement are challenges to be addressed. Predicting behavior and possible loads in advance and training the controller functions can be of help for these challenges. In the dynamic service-oriented fog layer, considering the computational load, connected forwarding devices and new fog region demand, the number of controllers can be changed, which may trigger the synchronization overhead. Further, using a bottom-up hierarchical approach of controller connectivity in the fog network layer, aggregating the fresh topology, the statistics at the main root or cloud controller by underlying controllers (domain, edge) can be prolonged. Similarly, hierarchical controller placement in a hybrid SDN fog infrastructure can have low performance for delay-sensitive applications. Other associated challenges of horizontally deploying the controller and servers at the fog layer are high energy consumption and carbon emission. Additionally, the controller collaboration model can have several dependent and independent control applications or processes, on the grounds that it can be arduous to debug the software multi-point applications inconsistencies. Moreover, it requires certain expertise by network administrators or configuration application developers.
- Fault identification in dynamic network topologies: In an SDN environment, the SDN controller decides on how to forward packets to make the network topology dynamic, which can be changed in real time, based on intents and flows, in some milliseconds. This hinders the correlation of faults and diagnosis of an SDN infrastructure, where it is needed to correlated information from the physical nodes but also the rules installed by the SDN controller. Nevertheless, some works tried to model dynamic network topologies in SDN, such as the multiagent distributed troubleshooting mechanism for SDN that identified faulty network links in the data plane impacting user experience

by Gheorghe et al. in [80], or the cross-layer self-diagnosis engine by Sanchez et al. in [81] to find the root cause and explain service outages in SDN, due to control or data plane faults.

- Trust and security: SDN poses some questions on the security and trust between the SDN applications and the controllers. Nevertheless, there are works that tried to improve trust among SDN components. Marconett et al. in [82], proposed a hierarchical broker-agent system to coordinate different SDN controllers to enhance the scalability of multi-domain SDN. Each broker is located at each domain to install flows on the data plane by means of the SDN controller of that domain, based on the concept of reputation to quantify the goodness of the flows installed by each broker. Betg-Brezetz et al. in [83] proposed a trust-oriented controller proxy that intermediates between the controllers and the data plane by making sure the flows sent by different controllers are correct. Other works focused on an efficient use of the networking resources by the SDN applications, such as that proposed by Isong et al. in [84], incorporating the notion of trust between SDN applications and the controller.
- Efficiency of SDN: The efficency of SDN should be improved. Huge amounts of computational overhead resulting from rules enforcement, overlapping network rules and the memory-restricted capacities of the OpenFlow-enabled devices are some of the reasons for low efficiency of SDN in hybrid cloud-fog systems. Computational context aware management in the control plane can improve SDN efficiency.
- ML and SDN in cloud-fog systems: The logical centralization of SDN has a crucial advantage, such as the capability to extract metrics from the SDN controller and the SDN underlying network elements. This allows to apply ML and artificial intelligence techniques to improve the efficiency and reliability of SDN through the available real-time information extracted from the data and control plane. One example is through intelligent knowledge extractors [85]. Another example is through knowledge-defined networking (KDN) [86], which is an architectural framework to evaluate network performance in real time through different types of ML techniques.
- Integration with other technologies for cloud-fog systems: The SDN-based cloud-fog deployment model can involve other technologies, such as blockchain [87] and multi-agent systems [88]. Blockchain can be used in the connection of controllers in the fog layer, which improves security issues. In addition, benefiting from the distributed nature of blockchain and multi-agent system, data integrity is ensured. The combination of these technologies with SDN needs to be studied and tested from different aspects.
- Consistency of the control channel: Some of the above-mentioned challenges are given for special type of controller design. Therefore, depending on the type of the controller design that is used, the challenges are different. Having a single administrator for enterprises and data centers, a centralized controller suits them, while having a heterogeneous environment, distributed and, of course, for now, hybrid controllers/T-model work better for cloud-fog systems [89].

  Maintaining the control channel consistency in a distributed control model (i.e., DSF [69], HECSDN [90]) is more challenging. Specifically, in a hierarchical control model, edge and domain controllers have to deal with security, mobility, and computation offloading. These local edge control policies are continually in transition and have unrest configurations in forwarding devices connected with fog nodes. Additionally, local edge policies and configuration should be inclined with global controller (residing at a cloud node or the fog top layer) forwarding policies. In the case of global policies change, the edge and domain controller suffer more delay because they have to reconfigure the fog layer forwarding device, according to global policies. Network reconfigurations, in the case of policy change, majorly include the flow forwarding rule addition, deletion, modification, barrier requests, and capabilities investigation to test the switches interoperability with fresh global policies and various other symmetric communications. In parallel, the policy change implementation can have the further challenge of forwarding rule overlapping, which invites policy violation at edge interfaces. Mudassar et al. in [91] addressed the policy conflict challenge, using a graph matching algorithm in the centralized SDN controller. Another overhead is flow rule debugging; whenever a policy composer or a semantic translator application has uncertainties, a single global control layer can initiate the application debugging mechanism in edge and fog controllers. Therefore, distributed controllers in fog require the intelligent mechanism for fault-free policy installation or abrogation in underlying fog layers to meet the stringent QoS constraints.

- Optical/Wireless networks and SDN in cloud-fog systems: So far, the benefits of SDN are mostly limited to wired networks [15]. Software defined wireless networking (SDWN) is an attempt to adapt SDN to wireless networks. Additionally, the software-defined optical network (SDON) emerged to enhance OpenFlow to support forwarding planes that are not capable of packet switching. SDWN and SDON need more attention to prepare cloud-fog systems for 5G and beyond.

7. Conclusions

By increasing the number of connected devices to the internet, high communication and computation capacities will be required to meet the demands of future applications (e.g., the massive IoT-related applications) requiring data sharing and processing. This paper discussed key enablers (i.e., future generation cellular technologies, cloud-related technologies, and computer network with softwarization) to support these future applications. We mainly focused on the opportunities and challenges of integrating SDN in cloud-fog systems for supporting 5G and beyond-enabled future applications. To this end, we first conducted a survey on current and emerging applications that use SDN in cloud-fog systems, where we found that most of them also considered 5G/6G technologies as another key enabler. Having experience of studying the emerging applications, this paper listed and discussed the potential challenges of using SDN in cloud-fog systems for 5G and beyond-enabled applications.

Author Contributions: Conceptualization and methodology, all authors; emerging cloud technologies, E.A. and S.A.; 5G and beyond, J.M.S.V., E.A. and S.A.; SDN, E.A.; role of SDN, S.M.R., E.A. and S.A.; SDN challenges, J.M.S.V., S.M.R. and S.A.; review and supervision, G.M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Statista: Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2030. Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 30 May 2021).
2. Malik, U.M.; Javed, M.A.; Zeadally, S.; Islam, S.U. Energy efficient Fog computing for 6G enabled massive IoT: Recent trends and future opportunities. *IEEE Internet Things J.* **2021**. [CrossRef]
3. Zhao, Y.; Wang, W.; Li, Y.; Meixner, C.C.; Tornatore, M.; Zhang, J. Edge Computing and Networking: A Survey on Infrastructures and Applications. *IEEE Access* **2019**, *7*, 101213–101230. [CrossRef]
4. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 13–17 August 2012.
5. Ahmad, M.A.; Patra, S.S.; Barik, R.K. Energy-Efficient Resource Scheduling in Fog Computing Using SDN Framework. In *Progress in Computing, Analytics and Networking. Advances in Intelligent Systems and Computing, 1119*; Das, H., Pattnaik, P., Rautaray, S., Li, K.C., Eds.; Springer: Singapore, 2020.
6. Gupta, H.; Dastjerdi, A.V.; Ghosh, S.K.; Buyya, R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and FC environments. *Softw. Pract. Exp.* **2017**, *47*, 1275–1296. [CrossRef]
7. Ijaz, A.; Zhang, L.; Grau, M.; Mohamed, A.; Vural, S.; Quddus, A.U.; Tafazolli, R. Enabling Massive IoT in 5G and Beyond Systems: PHY Radio Frame Design Considerations. *IEEE Access* **2016**, *4*, 3322–3339. [CrossRef]
8. Pärssinen, A.; Alouini, M.; Berg, M.; Kuerner, T.; Kyösti, P.; Leinonen, M.E.; Matinmikko-Blue, M.; McCune, E.; Pfeiffer, U.; Wambacq, P. (Eds.) White Paper on RF Enabling 6G—Opportunities and Challenges from Technology to Spectrum. (6G Research Visions, No. 13). University of Oulu. 2020. Available online: http://urn.fi/urn:isbn:9789526228419 (accessed on 30 May 2021).

9. Son, J.; Buyya, R. A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef]
10. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything*; Springer: Singapore, 2018; pp. 103–130.
11. Al-Ansi, A.; Al-Ansi, A.M.; Muthanna, A.; Elgendy, I.A.; Koucheryavy, A. Survey on Intelligence Edge Computing in 6G: Characteristics, Challenges, Potential Use Cases, and Market Drivers. *Future Internet* **2021**, *13*, 118. [CrossRef]
12. Ahvar, E.; Orgerie, A.C.; Lebre, A. Estimating Energy Consumption of Cloud, Fog and Edge Computing Infrastructures. *IEEE Trans. Sustain. Comput.* **2019**. [CrossRef]
13. Copeland, R.; Copeland, M.; Ahvar, S.; Crespi, N.; Shagdar, O.; Durand, R. Automotive virtual edge communicator (AVEC) with vehicular inter-agent service orchestration and resourcing (ViSOR). *Ann. Telecommun.* **2019**, *74*, 655–662. [CrossRef]
14. Sedaghat, S.; Jahangir, A. RT-TelSurg: Real Time Telesurgery using SDN, Fog, and Cloud as infrastructures. *IEEE Access* **2021**, *9*, 52238–52251. [CrossRef]
15. Baktir, A.C.; Ozgovde, A.; Ersoy, C. How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2359–2391. [CrossRef]
16. Mouradian, C.; Kianpisheh, S.; Abu-Lebdeh, M.; Ebrahimnezhad, F.; Jahromi, N.T.; Glitho, R.H. Application component placement in NFV-based hybrid cloud/fog systems with mobile fog nodes. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1130–1143. [CrossRef]
17. Aburukba, R.O.; AliKarrar, M.; Landolsi, T.; El-Fakih, K. Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing. *Future Gener. Comput. Syst.* **2020**, *111*, 539–551. [CrossRef]
18. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J.P. All one needs to know about Fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [CrossRef]
19. Yousefpour, A.; Ishigaki, G.; Gour, R.; Jue, J.P. On reducing iot service delayvia Fog offloading. *IEEE Internet Things J.* **2018**, *5*, 998–1010. [CrossRef]
20. Sarkar, S.; Chatterjee, S.; Misra, S. Assessment of thesuitability of Fog computing in the context of internet of things. *IEEE Trans. Cloud Comput.* **2018**, *6*, 46–59. [CrossRef]
21. OpenFogConsortium. OpenFog Reference Architecture for Fog Computing. 2017. Available online: https://www.openFogconsortium.org/ra/ (accessed on 30 May 2021).
22. IEEE P1931.1-Roof Computing, Framework and Use Case Scenarios. Available online: https://standards.ieee.org/develop/project/1931.1.html (accessed on 30 May 2021).
23. Meloni, A.; Madanapalli, S.; Divakaran, S.K.; Browdy, S.F.; Paranthaman, A.; Jasti, A.; Kumar, D. Exploiting the IoT Potential of Blockchain in the IEEE P1931.1 ROOF Standard. *IEEE Commun. Stand. Mag.* **2018**, *2*, 38–44. [CrossRef]
24. Ahammad, I.; Khan, M.A.R.; Salehin, Z. Software-Defined Dew, Roof, Fog and Cloud (SD-DRFC) Framework for IoT Ecosystem: The Journey, Novel Framework Architecture, Simulation, and Use Cases. *SN Comput. Sci.* **2021**, *2*, 159. [CrossRef]
25. NGMN Alliance. 5G White Paper. Next Generation Mobile Networks. White Paper. 2015. Available online: https://www.ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf (accessed on 30 May 2021).
26. Dogra, A.; Jha, R.K.; Jain, S. A Survey on Beyond 5G Network With the Advent of 6G: Architecture and Emerging Technologies. *IEEE Access* **2021**, *9*, 67512–67547. [CrossRef]
27. Piran, M.J.; Suh, D.Y. Learning-Driven Wireless Communications, towards 6G. In Proceedings of the 2019 International Conference on Computing, Electronics and Communications Engineering (iCCECE), London, UK, 22–23 August 2019; pp. 219–224.
28. Chowdhury, M.; Shahjalal, M.; Shakil, A.; Min, J.Y. 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions. *IEEE Open J. Commun. Soc.* **2020**, *1*, 957–975. [CrossRef]
29. Chen, S.; Liang, Y.C.; Sun, S.; Kang, S.; Cheng, W.; Peng, M. Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed. *IEEE Wirel. Commun.* **2020**, *27*, 218–228. [CrossRef]
30. GSMA. Introduction to Network Slicing. Available online: https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf (accessed on 30 May 2021).
31. Foukas, X.; Patounas, G.; Elmokashfi, A.; Marina, M.K. Network Slicing in 5G: Survey and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 94–100. [CrossRef]
32. Network Functions Virtualisation (NFV); Architectural Framework. 2013. Available online: https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf (accessed on 30 May 2021).
33. Velasquez, K.; Abreu, D.P.; Goncalves, D.; Bittencourt, L.; Curado, M.; Monteiro, E.; Madeira, E. Service Orchestration in Fog Environments. In Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic, 21–23 August 2017; pp. 329–336.
34. Ordonez-Lucena, J.; Ameigeiras, P.; Lopez, D.; Ramos-Munoz, J.J.; Lorca, J.; Folgueira, J. Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 80–87. [CrossRef]
35. Software-Defined Networking: The New Norm for Networks. Open Networking Foundation, White Paper. 2012. Available online: https://www.opennetworking.org (accessed on 30 May 2021).
36. Yuanzhe, L.; Jie, H.; Qibo, S.; Tao, S.; Shangguang, W. Cognitive Service Architecture for 6G Core Network. *IEEE Trans. Ind. Inform.* **2021**. [CrossRef]

37. Rodoshi, R.T.; Kim, T.; Choi, W. Resource Management in Cloud Radio Access Network: Conventional and New Approaches. *Sensors* **2020**, *9*, 2708. [CrossRef]
38. Habibi, M.A.; Nasimi, M.; Han, B.; Schotten, H.D. A Comprehensive Survey of RAN Architectures Toward 5G Mobile Communication System. *IEEE Access* **2019**, *7*, 70371–70421. [CrossRef]
39. Bouras, C.; Ntarzanos, P.; Papazois, A. Cost modeling for SDN/NFV based mobile 5G networks. In Proceedings of the 2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Lisbon, Portugal, 18–20 October 2016; pp. 56–61.
40. Kreutz, D.; Ramos, F.M.V.; Verssimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* **2014**, *103*, 14–76. [CrossRef]
41. Sezer, S.; Scott-Hayward, S.; Chouhan, P.K.; Fraser, B.; Lake, D.; Finnegan, J.; Viljoen, N.; Miller, M.; Rao, N. Are We Ready for SDN? Implementation Challenges for Software-Defined Networks. *IEEE Commun. Mag.* **2013**, *51*, 36–43. [CrossRef]
42. Hamed, M.I.; ElHalawany, B.M.; Fouda, M.M.; Eldien, A.S.T. A novel approach for resource utilization and management in SDN. In Proceedings of the 13th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 27–28 December 2017.
43. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [CrossRef]
44. Vilalta, R.; Via, S.; Mira, F.; Casellas, R.; Muñoz, R.; Alonso-Zarate, J.; Kousaridas, A.; Dillinger, M. Control and Management of a Connected Car Using SDN/NFV, Fog Computing and YANG data models. In Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 25–29 June 2018; pp. 378–383.
45. Margariti, S.V.; Dimakopoulos, V.V.; Tsoumanis, G. Modeling and Simulation Tools for Fog Computing—A Comprehensive Survey from a Cost Perspective. *Future Internet* **2020**, *12*, 89. [CrossRef]
46. Wang, T.; Qiu, L.; Sangaiah, A.K.; Xu, G.; Liu, A. Energy efficient and trustworthy data collection protocol based on mobile fog computing in internet of things. *IEEE Trans. Ind.* **2020**, *16*, 3531–3539. [CrossRef]
47. Valiveti, H.B.; Duggineni, C. Software Defined Device to Device Communication Handover- Latest Advancements. In Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 1079–1083.
48. Phan, L.A.; Nguyen, D.T.; Lee, M.; Park, D.H.; Kim, T. Dynamic Fog-to-Fog offloading in SDN-based Fog computing systems. *Future Gener. Comput. Syst.* **2021**, *117*, 486–497. [CrossRef]
49. Hou, X. Reliable Computation Offloading for Edge-Computing-Enabled Software-Defined IoV. *IEEE Internet Things J.* **2020**, *7*, 7097–7111. [CrossRef]
50. Lian, T.; Zhou, Y.; Wang, X.; Cheng, N.; Lu, N. Predictive Task Migration Modeling in Software Defined Vehicular Networks. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019; pp. 570–574.
51. Cao, B.; Sun, Z.; Zhang, J.; Gu, Y. Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3832–3840. [CrossRef]
52. Storck, C.R.; Duarte-Figueiredo, F. A 5G V2X Ecosystem Providing Internet of Vehicles. *Sensors* **2019**, *19*, 550. s19030550. [CrossRef]
53. Kiran, N.; Pan, C.; Wang, S.; Yin, C. Joint resource allocation and computation offloading in mobile edge computing for SDN based wireless networks. *J. Commun. Netw.* **2019**, *22*, 1–11. [CrossRef]
54. Ahammad, I.; Khan, M.A.R.; Salehin, Z.U. QoS Performance Enhancement Policy through Combining Fog and SDN. *Simul. Model. Pract. Theory* **2021**, *109*, 102292. [CrossRef]
55. Ateya, A.A.; Algarni, A.D.; Hamdi, M.; Koucheryavy, A.; Soliman, N.F. Enabling Heterogeneous IoT Networks over 5G Networks with Ultra-Dense Deployment—Using MEC/SDN. *Electronics* **2021**, *10*, 910. [CrossRef]
56. Son, J.; Dastjerdi, A.V.; Calheiros, R.N.; Ji, X.; Yoon, Y.; Buyya, R. CloudSimSDN: Modeling and Simulation of Software-Defined Cloud Data Centers. In Proceedings of the 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, China, 4–7 May 2015; pp. 475–484. [CrossRef]
57. Javanmardi, S.; Shojafar, M.; Mohammadi, R.; Nazari, A.; Persico, V.; Pescapè, A. FUPE: A Security Driven Task Scheduling Approach for SDN-based IoT-Fog Networks. *J. Inf. Secur. Appl.* **2021**, *60*, 102853.
58. Aujla, G.S.; Chaudhary, R.; Kaur, K.; Garg, S.; Kumar, N.; Ranjan, R. SAFE: SDN-assisted framework for edge–Cloud interplay in secure healthcare ecosystem. *IEEE Trans. Ind. Inform.* **2018**, *15*, 469–480. [CrossRef]
59. Tang, W.; Zhang, K.; Zhang, D.; Ren, J.; Zhang, Y.; Shen, X. Fog-enabled smart health: Toward cooperative and secure healthcare service provision. *IEEE Commun. Mag.* **2019**, *57*, 42–48. [CrossRef]
60. Li, J.; Cai, J.; Khan, F.; Rehman, A.U.; Balasubramaniam, V.; Sun, J.; Venu, P. A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System. *IEEE Access* **2020**, *8*, 135479–135490. [CrossRef]
61. Moyano, R.F.; Fernández, D.; Merayo, N.; Lentisco, C.M.; Cárdenas, A. NFV and SDN-Based Differentiated Traffic Treatment for Residential Networks. *IEEE Access* **2020**, *8*, 34038–34055. [CrossRef]
62. Chekired, D.A.; Togou, M.A.; Khoukhi, L. A Hybrid SDN Path Computation for Scaling Data Centers Networks. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [CrossRef]

63. Ibrar, M.; Wang, L.; Muntean, G.M.; Chen, J.; Shah, N.; Akbar, A. IHSF: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems. *IEEE Internet Things J.* **2020**. [CrossRef]
64. Akbar, A.; Ibrar, M.; Jan, M.A.; Bashir, A.K.; Wang, L. SDN-enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multi-Objective Optimization. *IEEE Internet Things J.* **2020**. [CrossRef]
65. Selvi, K.T.; Thamilselvan, R. Dynamic Resource Allocation for SDN and Edge Computing based 5G Network. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 19–22.
66. Dong, L.; da Fonseca, N.L.S.; Zhu, Z. Application-driven Provisioning of Service Function Chains over Heterogeneous NFV Platforms. *IEEE Trans. Netw. Serv. Manag.* **2020**. [CrossRef]
67. Li, B.; Deng, X.; Deng, Y. Mobile-edge computing-based delay minimization controller placement in SDN-IoV. *Comput. Netw.* **2021**, *193*, 108049. [CrossRef]
68. Altekar, G.; Stoica, I. *Electrical Engineering and Computer Sciences*; University of California at Berkeley: Berkeley, CA, USA, 2010.
69. Almadani, B.; Beg, A.; Mahmoud, A. DSF: A Distributed SDN Control Plane Framework for the East/West Interface. *IEEE Access* **2021**, *9*, 26735–26754. [CrossRef]
70. ETSI NFV Group Specification Draft. Network Functions Virtualisation (NFV). Ecosystem. Report on SDN Usage in NFV Architectural Framework. 2015. Available online: http://www.etsi.org/deliver/etsigs/NFV-EVE/001099/005/01.01.0160/gsNFV-EVE005v010101p.pdf (accessed on 30 May 2021).
71. Khorsandroo, S.; Sánchez, A.G.; Tosun, A.S.; Rodríguez, J.M.A.; Doriguzzi-Corin, R. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. *Comput. Netw.* **2021**, *192*, 107981. [CrossRef]
72. Palattella, M.R.; Soua, R.; Khelil, A.; Engel, T. Fog computing as the key for seamless connectivity handover in future vehicular networks. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), Limassol, Cyprus, 8–12 April 2019.
73. da Silva, A.S.; Smith, P.; Mauthe, A.; Filho, A.E.S. Resilience support in software-defined networking: A survey. *Comput. Netw.* **2015**, *92*, 189–207. [CrossRef]
74. Fonseca, P.; Bennesby, R.; Mota, E.; Passito, A. Resilience of SDNs based On active and passive replication mechanisms. In Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013; pp. 2188–2193.
75. Sharma, S.; Staessens, D.; Colle, D.; Pickavet, M.; Demeester, P. Fast failure recovery for in-band OpenFlow networks. In Proceedings of the 2013 9th International Conference on the Design of Reliable Communication Networks (DRCN), Budapest, Hungary, 4–7 March 2013; pp. 52–59.
76. Heller, B.; Sherwood, R.; McKeown, N. The controller placement problem. In Proceedings of the First Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, 13 August 2012.
77. Müller, L.F.; Oliveira, R.R.; Luizelli, M.C.; Gaspary, L.P.; Barcellos, M.P. Survivor: An enhanced controller placement strategy for improving SDN survivability. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 1909–1915.
78. Ros, F.J.; Ruiz, P.M. Five nines of southbound reliability in software-defined networks. In Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, Chicago, IL, USA, 22 August 2014.
79. Shirmarz, A.; Ghaffari, A. Taxonomy of controller placement problem (C) optimization in Software Defined Network (SDN): A survey. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–26. [CrossRef]
80. Gheorghe, G.; Avanesov, T.; Palattella, M.; Engel, T.; Popoviciu, C. SDN-RADAR: Network troubleshooting combining user experience and SDN capabilities. In Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, UK, 13–17 April 2015; pp. 1–5.
81. Sánchez, J.M.; Yahia, I.G.B.; Crespi, N. Self-modeling based diagnosis of services over programmable networks. In Proceedings of the 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, Korea, 6–10 June 2016; pp. 277–285.
82. Marconett, D.; Yoo, S.J.B. FlowBroker: A Software-Defined Network Controller Architecture for Multi-Domain Brokering and Reputation. *J. Netw. Syst. Manag.* **2015**, *23*, 328–359. [CrossRef]
83. Betgé-Brezetz, S.; Kamga, G.; Tazi, M. Trust support for SDN controllers and virtualized network applications. In Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, UK, 13–17 April 2015; pp. 1–5.
84. Isong, B.; Kgogo, T.; Lugayizi, F.; Kankuzi, B. Trust establishment framework between SDN controller and applications. In Proceedings of the 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Kanazawa, Japan, 26–28 June 2017; pp. 101–107.
85. Li, Z.; Zhao, Y.; Li, Y.; Rahman, S.; Wang, F.; Xin, X.; Zhang, J. Fault Localization based on Knowledge Graph in Software-Defined Optical Networks. *J. Light. Technol.* **2021**. [CrossRef]
86. Mestres, A.; Rodriguez-Natal, A.; Carner, J.; Barlet-Ros, P.; Alarcón, E.; Solé, M.; Muntés-Mulero, V.; Meyer, D.; Barkai, S.; Hibbett, M.J.; et al. Knowledge-Defined Networking. *SIGCOMM Comput. Commun. Rev.* **2017**, *47*, 2–10. [CrossRef]
87. Sharma, P.K.; Chen, M.; Park, J.H. A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT. *IEEE Access* **2018**, *6*, 115–124. [CrossRef]
88. Gharbi, C.; Hsairi, L.; Zagrouba, E. A Secure Integrated Fog Cloud-IoT Architecture Based on Multi-Agents System and Blockchain. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence, Vienna, Austria, 4–6 February 2021.

89. Ahmad, S.; Mir, A.H. Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers. *J. Netw. Syst. Manag.* **2021**, *29*, 9. [CrossRef]
90. Lin, F.P.C.; Tsai, Z. Hierarchical edge-Cloud SDN controller system with optimal adaptive resource allocation for load-balancing. *IEEE Syst. J.* **2019**, *14*, 265–276. [CrossRef]
91. Hussain, M.; Shah, N.; Tahir, A. Graph-based policy change detection and implementation in SDN. *Electronics* **2019**, *8*, 1136. [CrossRef]