A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV

IQBAL ALAM, KASHIF SHARIF, FAN LI, ZOHAIB LATIF, M. M. KARIM, SUJIT BISWAS, and BOUBAKR NOUR, Beijing Institute of Technology, China
YU WANG, Temple University, USA

Internet of Things (IoT) and Network Softwarization are fast becoming core technologies of information systems and network management for the next-generation Internet. The deployment and applications of IoT range from smart cities to urban computing and from ubiquitous healthcare to tactile Internet. For this reason, the physical infrastructure of heterogeneous network systems has become more complicated and thus requires efficient and dynamic solutions for management, configuration, and flow scheduling. Network softwarization in the form of Software Defined Networks and Network Function Virtualization has been extensively researched for IoT in the recent past. In this article, we present a systematic and comprehensive review of virtualization techniques explicitly designed for IoT networks. We have classified the literature into software-defined networks designed for IoT, function virtualization for IoT networks, and software-defined IoT networks. These categories are further divided into works that present architectural, security, and management solutions. Besides, the article highlights several short-term and long-term research challenges and open issues related to the adoption of software-defined Internet of Things.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Networks** → **Network structure**; **Network manageability**; **Programmable networks**;

Additional Key Words and Phrases: Internet of Things, network softwarization, software-defined network, network function virtualization, software-defined IoT

ACM Reference format:
Iqbal Alam, Kashif Sharif, Fan Li, Zohaib Latif, M. M. Karim, Sujit Biswas, Boubakr Nour, and Yu Wang. 2020. A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV. *ACM Comput. Surv.* 53, 2, Article 35 (April 2020), 40 pages.
https://doi.org/10.1145/3379444

1 INTRODUCTION

The Internet of Things (IoT) [103] enables connectivity of anything from anywhere at any time, creating ubiquitous and autonomous networks of heterogeneous devices. The IoT is tightly coupled

The work of F. Li is funded by National Natural Science Foundation of China No. 61772077 and No. 61370192 and Beijing Natural Science Foundation No. 4192051.

Authors' addresses: I. Alam, K. Sharif (corresponding author), F. Li (corresponding author), Z. Latif, M. M. Karim, S. Biswas, and B. Nour, Beijing Institute of Technology, Dept. of Computer Science, 5 S. Zhongguancun Street, Haidian Dist., Beijing, 100081, China; emails: {iqbalalam, kashif, fli, z.latif, mkarim, sujitedu}@bit.edu.cn, n.boubakr@ieee.org; Y. Wang, Temple University, Dept. of Computer and Information Sciences, 1801 N. Broad Street, Philadelphia, PA, 19122, USA; email: wangyu@temple.edu.

0360-0300/2020/04-ART35 $15.00
https://doi.org/10.1145/3379444

with numerous other technologies and devices, such as laptops, smartphones, home appliances, industrial systems, e-health devices, surveillance equipment, precision farming sensors, and other accessories connected to the Internet, which are expected to exceed 45 billion by 2020 with a financial share of USD 14 billion [18, 41]. This multitude of devices will produce large volumes of data; hence, the need for installing new network access and core devices will also proportionally increase. For smooth functionality and integration of such large-scale IoT systems, numerous technological challenges exist, such as security, privacy, and heterogeneity of devices as well as applications, edge access, and topological structure of IoT nodes, communication protocols, and data collection and analysis. Moreover, the potential mobility of IoT devices creates dynamic topological changes that may require re-establishment of flows. The application diversity at the top adds to the complexity of the ecosystem. It is perhaps impossible to create a unified solution for all the challenges; however, virtualization and programmability of hardware and software resources can significantly reduce the complexity of individual solutions.

Virtualization is the logical abstraction of the underlying hardware devices within a network, through software implementation. This abstraction decouples the control from hardware and makes it easier to modify, manage, and upgrade. In recent times, the abstraction has not been limited to hardware only, but rather software embedded into hardware has also been virtualized as independent virtual function elements.

Traditional networks are usually rigid and fixed. Heterogeneity, scalability, and interoperability have been major challenges due to the rapid growth of the Internet. Software-Defined Networks (SDN) [92] and Network Function Virtualization (NFV) [73] are two basic solutions for virtualization in communication networks. SDN creates a programmable network by centralizing the control functions of routing devices; hence, the physical infrastructure only acts as a forwarding (data) plane, while the controller dictates flow and policy information. OpenFlow (OF) [83] is the foundational protocol for interaction between the control and data plane, although other solutions such as in References [33, 45, 106] also provide similar functionalities. The main advantage of SDN is the single-point programmability allowing networkwide policy, flow, and configuration enforcement. Additionally, optimization of resource usage, virtualization of network slices, security, and vendor independence can be achieved. NFV is the mechanism of abstracting functions, such as firewall, load balancing, path calculation, and so on, from dedicated hardware to a virtual environment. The key benefits of NFV include replacing dedicated hardware with commodity servers. It enables SDN applications like security functions, load balancing, data collection and analysis, and so on, through the deployment of on-demand virtual network functions (VNFs). This allows not only enhanced scalability and elasticity for deploying vendor-independent commodities with reduced cost but also optimizes computing, memory, storage, and networking capacity of network devices. SDN and NFV are not competing for solutions (even though they are maintained by different standardization organizations); rather, they complement each other. Hence, the key benefits of both technologies are inter-related. NFV can boost SDN toward virtualizing the SDN controller and other network applications in the cloud. Similarly, SDN with its programmable network connectivity can implement traffic engineering decisions taken by VNFs [65].

The use of SDN along with IoT has been studied in some detail. Several solutions have been proposed to address different IoT optimization challenges by using software-defined networking. Similarly, Network Functions [24] of IoT devices and ecosystems can also be virtualized to make them more agile, robust, and cost-effective. This will reduce the number of physical devices needed, easily segment networks, and enforce security policies on physical devices. Based on these challenges, the solutions presented in the literature have attempted to solve different issues mostly in isolation. Due to a lack of standardization efforts specifically for virtualized IoT device, IoT controllers, and controller to device interfaces, we argue that there is a need for a comprehensive

Table 1. Existing Surveys and Contribution of This Work

Survey	Year	Main Focus	Details
Farris et al. [36]	2019	Addresses security features provided by SDN and NFV for IoT.	Detailed discussion about IoT protection, monitoring, and reaction to security threats. Virtualization and software-definition for IoT is not the main focus. Review covers 2000–2018.
Pan et al. [85]	2018	IoT applications based on edge, cloud, and edge computing.	Brief introduction of challenges and enabling cloud-based technologies for IoT applications: NFV and SDN. Review covers 2009–2016.
Akpakwu et al. [2]	2018	5G for IoT: Communication technologies and challenges.	Limited to IoT application use cases for mobile communications. Briefly introduces SDN and NFV technologies. Review covers 2002–2017.
Cox et al. [25]	2017	SDN advancement survey.	Discusses SDN state of art and challenges. Brief discussion on SDN-IoT, NFV, and SDIoT. Review covers 2002–2016.
Ngu et al. [77]	2017	IoT Middleware issues and enabling technologies.	Focuses on middleware with limited discussion on virtualization. Review covers 2003–2016.
Bizanis et al. [15]	2016	SDN and virtualization for IoT.	Focuses on SDN and NV in IoT applications, specifically in mobile and cellular context and limited to 5G and WSN. Review covers 2009–2016.
Khan et al. [57]	2016	WSN virtualization.	Limited to detailed discussion about WSN virtualization, state-of-the-art, and research issues. IoT is not the main focus. Review covers 2003–2016.
This work	2020	IoT virtualization using SDN, NFV, NV, and hybrid SD designs.	Discusses solutions that are specific to IoT. Literature is covered that utilizes software-defined networking (network layer), function virtualization, hypervisors, hybrid NFV and SDN, and software-defined Internet of Things. Review covers all literature until 2019.

survey of such solutions so that the research community can benefit from an in-depth analysis of existing works and research directions derived from it. Therefore, this work fills this gap between the existing work and future direction of softwarized IoT networks.

Virtualization, SDN, and IoT have individually attracted tremendous attention from the research community [16, 36, 50, 54, 86, 103, 107]. However, there has been very limited effort to review the literature that combines and presents a comparative analysis in a single document. Table 1 lists surveys that have previously been done and are related to this work. It is important to note that most of them only target a specific technology. The closest works are References [15, 36, 57], which deal with the virtualization in IoT and WSN. Bizanis et al. [15] provide a survey of literature from 2009 to 2016 that mostly focuses on SDN and network virtualization in IoT applications, specific to mobile, cellular, and 5G context. It does not cover IoT in depth nor does it consider all solutions available in the literature for the given time frame. Khan et al. [57] focus specifically on WSN and do not collect works on IoT in general. Farris et al. [36] present an excellent SDN- and NFV-based IoT survey, but it focuses on security.

The original contribution of our work is presented in six parts. First, we give background and fundamental information on different softwarization and virtualization techniques for network functions, devices, and IoT in Section 2. We also highlight the IoT issues that will benefit from virtualization. Second, we present a discussion and comparative analysis of software-defined networks used in the IoT system in Section 3. The third part, in Section 4, elaborates on the function virtualization of the IoT ecosystem and its augmentation with SDN. It is important to note that virtualization of the network layer is not the only solution available; hence the fourth part in Section 5 reviews works that can be implemented to virtualize other aspects of the system, such as configuration, management, data collections, and so on. In the fifth part, in Section 6, we present a collection of security solutions for IoT that is addressed by SDN-based, NFV-based, and SD-based techniques. Last, we elaborate on lessons learned and future research directions based on previous observations, in Section 7.

2 BACKGROUND

Network virtualization [24] is the mechanism of abstracting network hardware resources and embedded software functionality into a single logical entity, which is usually referred to as *virtual network*. In other words, successful network virtualization would require platform virtualization along with resource virtualization. Usually, this is achieved through a Virtualization Layer, which is an additional abstraction layer between network (and lower) layer devices and embedded applications in them. Virtualization can be categorized as (1) External, which groups multiple physical resources and presents it as a single virtual entity, or (2) Internal, where a software container on a single server presents a network like functionality.

2.1 Control Plane Virtualization

Traditionally, a network comprises of hardware devices for connectivity with a dedicated controller built into them. The controller is part of router architecture that instructs switches where to forward packets. A flexible and feature controller that can be remotely and securely configured can increase the efficiency of the whole network. This requirement has led to the virtualization of the controller, which is implemented through Software Defined Networks (SDNs) [25]. The objective is to split decision making and packet forwarding, i.e., the routing algorithms of router/switch are split from the packet-forwarding engine and placed in the control plane. This may be done centrally or in a distributed manner. The SDN controller supports programmability, allowing the data plane infrastructure to be abstracted for management layer applications and service. Programmability [37] refers to the ability to enhance network features, linking the applications to it and allowing dynamic traffic flow changes, providing both network- and application-level Quality of Service (QoS).

2.2 Function Virtualization

Function Virtualization is implemented through an NFV architecture, which utilizes different techniques to virtualize complete network node functions, into series of building blocks to establish connectivity and to create communication services among them. NFV design consists of three main components as follows [24]: (A) VNF: These are the software functionalities responsible for executing specific network operations; (B) Network Function Virtualization Infrastructure (NFVI): This framework manages different VNFs, virtual storage, and processing; and (C) Network Function Virtualization Management and Orchestration (NFV-MANO): It provides an architectural framework to interfaces and reference points of individual VNFs and NFVI elements.

2.3 Device Virtualization

Device virtualization is the process of virtualizing a switch in the data plane using logical abstractions of its components or functionality, which can be executed in the cloud. Virtualization, in a computing platform, tends to hide the physical features from the users and create an abstract computing platform to define unique rules for switches to comply, which may be referred to as VNFs. The software that controls virtualization is called the control program, also referred to as hypervisor [28]. Similarly, sensor virtualization [58] provides software abstraction of various IoT objects, which can be accessed by application through simple interfaces. Zeroconf [108] or similar APIs allow the virtual sensor to transparently discover arbitrary sensor device as virtual switches. It is also able to communicate with different applications using a standard communication interface of UDP/TCP sockets or HTTP [34]. Hence, the applications are not required to deal with hardware-specific details.

3 SOFTWARE-DEFINED NETWORK-BASED IOT

SDN-based IoT is a concept where virtualization of access network for IoT devices can provide routing efficiency, network management, and resource optimization for increasing the needs of IoT networks [109]. SDN solutions in the IoT environment are expected to resolve traditional network issues [120], like heterogeneity, interoperability, and scalability among IoT devices, inefficient service deployment (lack of dynamic services), slow adaptation to new services (network upgrade time consumption), and lack of user experience guarantees (minimum bandwidth). To do so, different SDN-based IoT architectures have been proposed in many works until recently. In addition to commercial solutions [49] a handful of proposals and solutions are available in academic literature. We classify them into architectural, security, and management solutions. SDN-based IoT architecture deals with clear separation of concern between services provided in the control plane and the data plane. The control plane specifies the management of network traffic and data plane specifies the mechanisms to forward traffic to the desired destination. SDN-based IoT management specifies how the applications on top of the Management Layer interact with the control plane and the coordination among them. It also allows the admin/analyst to define how the control process is to be governed not only by the SDN controller itself but also by human users. SDN-based IoT security specifies different security parameters for access to a network, end-point devices, and other control layer elements, which are elaborated in a separate section. In recent times, the focus has moved toward other aspects of virtualization as compared to the generic SDN concepts, as depicted in Figure 2.

3.1 Architecture Solutions

The solutions that address the architectural issues of SDN-based IoT can be viewed from several perspectives, such as (a) device connectivity (to the cloud/devices) through soft or hard gateways; (b) multi-IoT network connectivity; (c) approaches for IoT scalability, heterogeneity, and interoperability; (d) cloud/edge/fog connectivity; and (e) migration to SDN. It is important to note that most of the solutions fall under multiple subcategories. Figure 1 depicts the SDN-based IoT architecture, which can be used as a reference point. The basic addition to the data plane is the extension of IoT devices; however, these are non-OpenFlow compliant devices that may not be visible to the control plane. Table 2 presents the comparative analysis of these architectures (along with classification and limitations), and the discussion below elaborates on the salient features and limitations of each.

Desai et al. [29] propose a broad architecture where IoT device communication with cloud-based processing systems is enabled using SDN. The proposed management device structure is designed for several different applications, such as smart homes, temperature sensors, and so on. The IoT

Table 2. Comparison of Different SDN Architectures for IoT Networks

*Ref.	Objectives	Solutions	*Control Plane Arch.	Controller	Benefits	Limitations
[29]	Heterogeneity, Device comm.	OF-enabled mgt. device.	D	NOX, POX, ODL	OF-enabled mgt. device may make network simpler	Implementation of the device is future work.
[31] *Cl.*	Scalability and latency. Device to cloud comm.	Priority-based flow space management.	C	SDN controller	Mathematical sol. for eff. flow-space allocation in fog.	Single SDN controller only. No multi-domain allocation.
[64]	Heterogeneity, Interoperability, Scalability, Security.	IoT gateways and SDN switches. Distributed network OS.	D	SDN controller	Distributed OS with centralized control. Global view of distributed network.	Architecture only. No performance evaluation or implementation.
[66]	Heterogeneity, Interoperability, Latency, Scalability, Security.	SDN gateway/router. Distributed network OS.	D	POX	Cross-domain IoT devices discovery. Real time evaluation for latency.	Sink devices not OF compliant. Security mechanisms require further exploration.
[70]	Heterogeneity, Interoperability, Latency, Security.	Efficient routing path.	D	SDN controller	Reduce workload of SDN controller.	Routing algo. not given.
[78] *Cl.*	Latency, QoS, Overhead, Mobility.	Services on edge devices. Lightweight control mechanism.	D	SDN controller	Efficient P2P service abstraction for devices. Reduced signaling and data overhead. Efficient resource management.	Controller compatibility with the proposed architecture may become an issue.
[81] *Cl.*	Scalability, QoS, Reliability, Security	Apps. on top of SDN controller. Dynamic end-to-end comm. for cloud to IoT devices.	D	Ryu, OpenFlow	IoT device recognition, and policy enforcement. Apps uses specialized controller for traffic analysis. Real-time data collection and analysis.	Scenario-specific solution (smart cities). Sensor bound to single tenant, thus limited device virtualization.
[82] *Cl.*	Heterogeneity, Scalability, Mobility.	Replacement of traditional gateway with SDN gateway.	D	ONOS, ODL	Improved network eff. and agility. Intelligent routing and caching techniques.	Architecture only. Evaluation and implementation is future work.
[88]	Heterogeneity, Interoperability, Scalability , Security, QoS.	Centralized global view. Heterogeneous devices and data formats used. Adaptable network state.	C	Layered IoT controller	Minimized latency and optimized interoperability and scalability. Better flow scheduling.	Security and resource provisioning can be improved.
[95]	Single solution for: Scalability, Heterogeneity, QoS, Latency, Reliability, Security.	Centralized SDN control with decentralized data management. Multi-layered model. SD-gateways in fog with specialized algorithms.	D	SDN controller	Inter-controller communication. Intelligent fog nodes. Controller uses mgt. protocols (NetConf and Yang, OF-Config and extended OF). Unified application for comm.	Architecture only. Implementation is future work.

*D: Distributed, C: Centralized, Cl.: Cloud/edge/fog.

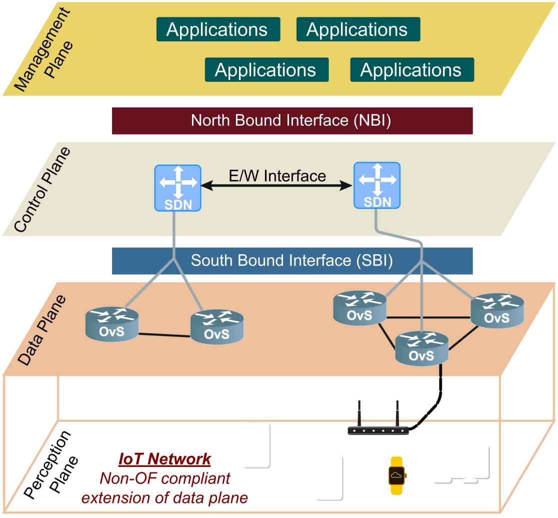


Fig. 1. A generic SDN-based architecture for IoT.

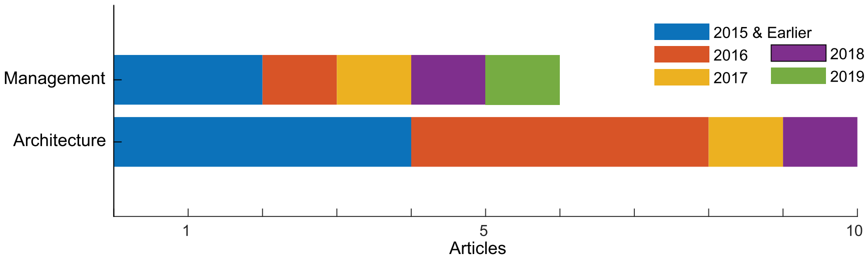


Fig. 2. Articles addressing the SDN-based IoT solutions in past five years

device collects the data that have to be processed, and the data manager module formats it appropriately for the application layer and forwards it to the OpenFlow-switch (OF-switch). The OF-switch works traditionally and consults the forwarding table for packet processing. Once the data reach a gateway controller, their processing location is determined by consultation with other gateway controllers. Data may be processed locally, which is trivial; however, for processing in the cloud they have to be sent to the cloud gateway controller. The complete management solution can be implemented inside a Linux kernel, and IoT devices can then be connected to such management devices. However, the authors have left the implementation of these open-flow-enabled management devices as future work. Similarly, Salman et al. [95] also propose an architecture with a layered model, for IoT device connectivity with decentralized data and centralized control. The proposed four-layered model consists of the Application, Control, Network, and Device Layers. The architecture uses unique identifiers in the device layer that ensure interoperability, security, and quick addresses. Software-Defined Gateways (SD-Gateways), a virtualized abstraction of a common gateway supporting extended OpenFlow protocol, is used to communicate with the SDN controllers. The SD-Gateway also acts as a fog node and bridges the communication using virtual functions. However, the implementation of SD-Gateways is not given. The Control Layer specifies the network orchestration and computation, such as collecting the topology data, defining security rules, and implementing scheduling algorithms. However, these algorithms have not been addressed in depth in the article and may be considered as future research directions. Using the virtual function concept, Li et al. [64] propose an SDN-based IoT architecture with conceptual virtual functions that has three layers. The Application Layer accommodates IoT services accessible through APIs. The Control Layer accommodates SDN controllers running a distributed OS, while the Infrastructure Layer comprises IoT gateways and SDN switches to enable connections between the SDN controller and IoT devices. The ideas presented also try to improve the scalability and heterogeneity of IoT devices as multiple technologies can be used. Moreover, the distributed OS improves Control Layer scalability. However, the work presented does not show a performance comparison or real-world implementation.

Ojo et al. [82] propose a replacement of traditional IoT gateways, with specialized SDN-enabled gateways. These gateways are capable of managing wired and wireless devices and are claimed to be more flexible, efficient, and scalable. The authors also claim that the gateway can perform efficient traffic engineering utilizing intelligent routing and caching techniques. However, the work is limited in defining intelligent routing algorithms and performance evaluation or implementation in real time, which is considered a future direction.

Li et al. [66] discusses issues like interoperability from the perspective of devices, data, communication protocols, and the re-usability of data generated from IoT devices. Moreover, the authors suggest resource utilization, openness, and interoperability by using a layered architecture that includes a Device Layer (for collecting data), a Communication Layer (contains SDN-enabled switches and gateways), a Computing Layer (with SDN Controller), and a Service Layer. The IoT devices communicate with the SDN gateway/router through sinks, like Raspberry Pi. An important limitation of this work is that the sink devices are not programmable and work independently. As all IoT devices send data to the sink, they can become overloaded. Keeping the same objectives of improving scalability and device heterogeneity, Martinez-Julia et al. [70] use an additional IoT Controller. The IoT controller acts as a functional block and receives communication interests by IoT agents installed on the objects, finds the responder in the network graph, determines the routing path, builds the forwarding rules for the object requested, and, finally, passes such rules to the SDN Controller for installation. The advantage is that the IoT Controller tends to reduce the workload of the SDN controller, but the limitations still may persist at the forwarding rule and routing step. The latency in discovering objects may also be present, as the author also stated that the IoT Controller may sometimes face protocol compatibility issues and hence some rules may need to be handled by the forwarders.

Nguyen et al. [78] present a distributed mobile edge-cloud architecture that enables a new network service abstraction called SDN-based IoT Mobile Edge Cloud Architecture (SIMECA). It aims at improving IoT device communication performance, as compared to the Long Term Evolution/Evolved Packet Core (LTE/EPC) architecture. It realizes the abstraction by lightweight control and data planes that significantly reduce signaling and packet header overhead, while supporting seamless mobility. Through evaluations, SIMECA shows promising improvements in data plane overhead, control plane latency, and end-to-end data plane latency, while coordinating large numbers of IoT devices in cellular networks. However, controller details are not given, which may impact the results if SDN controllers with different features are used. Other issues like heterogeneity, availability, and scalability may also exist from the device perspective. Another fog-based solution is proposed by Diro et al. [31]. The objective here is to optimize flow space allocation and reduce latency while processing using SDN-based fog computing dedicated to IoT application. The latency caused due to flow entry installation is mainly due to control space designs and critical data packets. Hence, the solution aims to mitigate these issues by isolating critical data packets and defining a customized priority flow class for IoT applications in the fog network. This also enhances the QoS capabilities of a heterogeneous IoT environment. Analytical results show that prioritized flow classes act more efficiently than normal flow classes. However, the solution only supports a centralized SDN controller, and multi-controller extension with distributed space allocation is a future work.

Table 3. Key Areas of SDN-based Architectural Solutions

Concentration Domains	**Literature** (D: Distributed, C: Centralized, Cl.: Cloud/edge/fog)
OF-enabled middleware	[29] (D), [64] (D), [66] (D), [82] (D, Cl)
Efficient routing	[31] (C, Cl), [70] (D), [78] (D, Cl), [88] (C)
Application/protocol-specific SDN-controller	[81] (D, Cl), [95] (D)

One step further from the device connectivity and individual network scalability/heterogeneity, Ogrodowczyk et al. [81] present an architecture that contains multiple independent IoT ecosystems connected through the cloud using SDN infrastructure. The solution can generate a global view of all IoT resources using OF Experimenter extensions. It also proposes a protocol that interfaces between the cloud orchestrator and the OF controller (a highly customized Ryu [6]). The solution is evaluated as the Poznan Smart City use case. The authors demonstrate the *slicing* of a city into different smart spaces, while connected to a single SDN-based platform. The citywide network is an OF-enabled infrastructure integrated with cloud resources, capable of hosting multi-tenant cloud applications for IoT devices. Most of the emulated analysis shows considerable improvements, but for real-time performance evaluation of a smart city and to scale the entire system, further testing is required to validate the feasibility of using vendor-independent sensor devices.

In contrast to proposals of new architectures, migration techniques of traditional IoT networks to SDN are also interesting. In this regard, Qin et al. [88] discusses a centralized Multi-network Information Architecture (MINA) to tackle the heterogeneity in IoT. It attempts to address the interoperability challenges with different heterogeneous devices and exploits various data formats for modeling information. MINA's objective is to minimize latency and optimize interoperability and scalability to improve QoS. The work uses an OpenFlow-like protocol, and the evaluations show that resource provisioning can be done effectively in IoT multi-network systems by using the observe-analyze-adopt loop [89]. It also defines flow scheduling over ad hoc heterogeneous paths and takes advantage of flow matching using heuristic algorithms to provide QoS. Its proposed flow scheduling algorithm shows significantly improved performance when compared to traditional solutions. However, there is very little work available in the literature related to migration architectures.

Insights: Several novel architectures have been proposed to tackle IoT challenges like heterogeneity, interoperability, latency, security, data manipulation, and so on. Table 3 shows the literature distribution in three key areas, where the majority of the solutions have a distributed approach. However, most works only present architectural details. Real-world implementation and experiments are needed to determine the true performance. Hence, this is a major research direction for this area. From the works discussed in sub-section, it is evident that more focus is on individual IoT device connectivity to the network or the cloud. One common thing in References [29, 64, 82, 95] is the use of gateways to connect the IoT device, where Reference [82] presents a solution for virtualizing these gateways. This is an interesting solution as the gateway in the IoT network may not be capable enough hence, virtualizing its functions can be helpful. This will further extend to multi-network solutions; however, the focus here is more on the connectivity in cloud/fog/edge elements and less on the IoT network itself. Furthermore, the adaption of existing controllers to IoT (migration strategies) is still not completely addressed. Controllers that can seamlessly integrate into access network and can reach devices in the mobile domain will be necessary to better optimize the IoT ecosystem.

3.2 Management Solutions

At the existing scale of deployed networks, it is almost impossible to manually configure remote devices. IoT requires that network providers can configure and reconfigure devices across the

Table 4. Comparison of SDN-based Solutions for IoT Management

*Ref.	Objectives	Solutions	Control Plane Arch.	Controller	Benefits	Limitations
[12]	Wireless extension. Real time mgt., Flexibility, Simplicity.	Device and Topology management.	Centralized	Customize controller	Application-aware service provisioning, and improved network performance.	Limited to specific sensor devices.
[21]	Congestion control, improve delays and jitters.	Comparative analysis of load balancing techniques. Device and Topology management.	Centralized	POX	Improved network performance.	Limited to POX controller only.
[38] *Cl.*	Heterogeneity, Scalability.	Management of data path across IoT, cloud, and edge network.	Distributed	ONOS	Congestion recovery with reliable data delivery.	Redirection of flows may create delays for time-sensitive mice flows.
[44]	Networking, Mobility, Standardization, Security, QoS.	IoT architecture combining SDN with message-based publish/ subscribe DDS middleware.	Centralized	SDN controller	Filtering and fusion mechanism for efficient traffic engineering.	Architecture design only. No implementation or evaluation.
[110] *Cl.*	Network load, Storage and Cost reduction, D2D communication.	Information filtering. Prioritization using VoI.	Centralized	SDN controller	Reduced load by information filtering.	Distributed and disruption tolerant architectures. Efficient information processing functions.
[117]	Network Slicing. Reduce latency, Efficient load balancing.	Slicing mechanism using Flowvisor for multiple home networks.	Centralized	NOX	Isolating network traffic and bandwidth. Resource sharing. Cost-effective.	Architecture lacks compatibility with all applications. Privacy, performance, security, and flexibility may be further improved.

*Cl.: Cloud/edge/fog.

network from a centralized management point. However, this requires the right technology to automate the whole management process. SDN can facilitate advanced mechanisms to configure and manage devices across a variety of different types of networks. Primarily the management solutions can be subdivided into (a) IoT application deployment, (b) device discovery/configuration in wireless and wide area networks, (c) Network slicing, and (d) cloud/fog/edge management. This section discusses different SDN-based IoT management solutions in the context listed above and Table 4 gives their comparative analysis.

Hakiri et al. [44] discuss five key network-related challenges of IoT, such as current standardization efforts, mobility management, recurring distributed systems issues, communication protocols, and security and privacy. They outline an IoT architecture that uses a message-based publish/subscribe Data Distribution Service (DDS) middleware address these by using SDN, where additional management services are also part of the network. Within a domain, DDS can provide discovery and communication service between different heterogeneous IoT devices and the controller itself. DDS is utilized in local network, whereas SDN is responsible for allowing the connection outside of a local network. A novel SDN-enabled gateway is also proposed for smooth handover migration between smart IoT devices in a Wide Area Network (WAN). However, this is only an architecture, and core algorithms of DDS are not defined.

In contrast to the above, Bera et al. [12] propose leveraging of IoT-related application-aware service in the Wireless Sensor Network environment. They present an architecture named Soft-WSN that is based on the centralized provisioning of the SDN controller. The architecture is divided into three layers. The application layer generates application-specific requests to be sent to the SDN controller. The control layer in addition to the SDN controller has two important entities to assist with policy management. The first one is the device manager, which deals with device-specific controls of scheduling the sensing tasks and active-sleep management. Second is the topology manager, which deals with network topology control mechanisms while focusing on the sensor network connectivity management and forwarding rules. The proposed system can be effective for several IoT applications, such as environmental and traffic monitoring, smart homes, and so on. From the experiment results, the authors show that the solution provides better data delivery rate, energy efficiency, and traffic overhead than traditional WSN. However, this may have compatibility issues with different radio technologies and controller placement problem in diverse networks.

SDNs can enable the installation and management of communications and computational resources to develop and deploy IoT applications. Sieve, Process, Forward (SPF), by Tortonesi et al. [110], is an extended SDN architecture of the Open Networking Foundation (ONF). The authors use SPF for information processing, replacement of data plane with Dissemination plane, and a novel SPF-Controller, with Programmable IoT Gateways (PIGs). It uses data processing solutions for audio/video analysis, IoT device discovery, and tracking at the edge of the network rather than in the cloud, which reduces high bandwidth usage. The SPF architecture has three stakeholders: administrators, service providers, and users. Administrators deploy, run, and operate SPF controllers along with PIGs, allowing the service providers to use it. Service providers develop, deploy, and manage IoT applications. Users may utilize the SPF applications available to them by installing their client app on their smart devices. This management solution can further be improved by incorporating IoT-specific application functions such as data aggregation and analysis with semantic understanding.

A real-time 5G Operating Platform proposed by Fichera et al. [38] can manage the heterogeneity and scalability of a network. A testbed has been presented in this work for exploiting SDN management capabilities to provide data delivery paths across different network domains under 5G communication. The experiment divides the testbed into IoT-based, cloud-based, and edge networks. A Service Orchestrator directs the cloud, SDN, and IoT Orchestrators, which then direct the respective resource infrastructure manager/controllers. Experimental results show that redirected operation takes less time, although packet dropping at congested switches may tend to degrade the real-time assured services of the proposed scheme.

Slicing techniques have always played a key role in securing and managing a complex network, especially for cloud/edge networks defined for different types of IoT systems. It allows the creation of multiple logical networks over a single physical infrastructure and provides efficiency, cost reduction, and flexibility. Technologies like SDN (through network programmability) and virtualization are the means to realize it. Slices may be optimized in many ways, including bandwidth and latency requirements. From the user's perspective, they only visualize a single network, even though it may physically be a portion of a layered network. Yiakoumis et al. [117] propose a prototype where multiple home networks can be sliced and a trustworthy third party can manage the whole network using different slicing techniques. Authors use FlowVisor [99] for a slicing mechanism in OpenFlow networks, providing bandwidth and traffic isolation. The solution also allows

Table 5. Key Areas of SDN-based Management Solutions

Concentration Domains	**Literature** (D: Distributed, C: Centralized, Cl.: Cloud/edge/fog)
Device and topology management.	[12] (C)
Load balancing	[21] (C), [110] (C, Cl), [38] (D, Cl), [44] (C)
Slicing mechanisms	[117] (C)

configuration of access points, firewalls, and NATs in smart home environments. The experimental evaluation also shows improved latency and load balancing performance. However, the proposed solution does not consider the virtualization of namespace and other resources, such as storage.

Wan et al. [112] integrates SDN and D2D communication into industrial process control to achieve dynamic management of IoT resource allocation, considering ontology modeling. SDN is utilized for intelligent data transmission and network control and to provide an abstraction of the underlying physical resources. Ontology modeling is used to access production-related information. Although this significantly improves the control, the use of cloud/edge within the industrial IoT setting could be beneficial.

Insights: Management of IoT networks or services for IoT networks based on SDN has not been explored to a great extent, as depicted by Table 5. Most of the management has been related to network optimization solutions (i.e., load balancing). As the primary job of a controller is network flow installation, less focus is given to management, although the SDN architecture has a complete management plane, where many different applications can be executed. Another important factor to note is that some of these applications can be run directly on top of the controller (almost as part of it). The best example of this is Reference [117], and it can be leveraged for large-scale IoT networks. Combining it with the architecture presented in Reference [81] and programmable gateway concepts of Reference [82], it can give a more comprehensive solution to the architectural and management challenges in programmable IoT networks. Some of the other directions worth exploring are synchronization and compatibility of IoT devices. APIs for such services can improve heterogeneity in the IoT ecosystem.

4 NETWORK FUNCTION VIRTUALIZATION FOR IOT

Network Function Virtualization and SDN are complementary technologies. They do not require or are dependent on each other but rather improve and facilitate each other's working. NFV provides a collection of virtual applications referred to as VNFs. These can include processes for deep packet inspection, routing, security, and traffic management, which can be combined to provide network services specialized for IoT [19]. A hybrid SDN/NFV architecture for IoT, given in Figure 3, shows a general interaction of SDN and NFV to provide reliable communication and to facilitate IoT platforms. The architecture is composed of (i) NFVI, which consists of the networking hardware and software resources required to connect and support carrier network; (ii) VNFs, which are responsible for managing specific network functionality that executes on one or multiple virtual machines (VMs); and (iii) the Management and Orchestration (MANO) plane, which facilitates connectivity among services of different modules of NFVI, VNF, and APIs from the Management Plane and coordinates with the respective sub-components. All these elements leverage each other to achieve sustainable network virtualization, with uninterrupted network connectivity and enforcing efficient packet flow rules by the SDN controller. The rest of the section presents architectural and management solutions for IoT using NFV, and Table 6 gives the comparative analysis, while Figure 4 shows the yearwise distribution of the articles reviewed in this section. It can be observed that recent years have more emphasis on integrating NFV with SDN solutions.

Table 6. Network Function Virtualization Solutions for IoT Networks

*Ref.	Objective(s)	Solution(s)	*Control Plane Arch.	Controller and Switch	Implementation, Evaluation, Benefits	Limitation(s)
[4] *a Cl.*	Min. cost and energy consumption. Software virtualization.	Cloud-based SDN and NFV solution for IoT infrastructure.	C	SDN controller	Multiple tenants can use the same solution simultaneously. Profit margin.	Third-party services can cause security threats. Resource optimization as future work.
[7] *m Cl.*	Security, scalability, flexibility, reusability, and congestion control.	Cloud-based Virtualization techniques for IoT devices to evolve SDaaS.	D	SDN controller	NFV infrastructure rendering OpenStack/Open Volcano APIs. Less devices and cost effective. Sleep mode for higher lifespan.	Software compatibility is very challenging.
[9] *m*	Efficient routing. Cost effective deployment.	Resolves CAPEX issues in IoT.	C	SDN controller	Efficient inter-domain routing. Less connected and deployed devices, hence cost-effective.	Latency
[32] *a s*	Security and privacy. Cost effective MVNO. Value-added services for MVNOs. Multi-MVNO networks.	Context-aware forwarding of IoT traffic. Contextual info. utilized.	C	Central service controller. IoT gateways. MVNO switch.	Cost effective business model for MVNO in IoT. Programmable MVNO IoT gateways. Trailer-slicing for IoT networks.	Proposed arch. may not be a unified IoT platform.
[64] *a s*	Routing, Access control, Security, Traffic control, Virtualization.	SDN-based IoT framework with NFV.	C	SDN controller and switches with IoT gateways.	Distributed OS. Performance, scalability, and security are enhanced due to virtualization.	Limited to the study of general arch.
[72] *a Cl.*	Heterogeneity, security, and reduce latency.	SDN/NFV solution supporting IIoT.	D	SDN controller	Cloud-based solutions. Embedded intelligence apps. Efficient routingand reduced latency.	Only supports 802.15.4. Network congestion is still challenging. 5G RAN extension in future.
[82] *a s Cl.*	Interoperability, Discovery, Scalability, Security, and mgt. flexibility. App.-specific requirement provisioning.	SDN-IoT arch. with NFV implementation.	—	SDN controller Virtualized IoT gateways	Enhanced performance and management of all resources. Device discovery with enhanced connectivity.	Scalability issues due to overloading of data traffic.
[96] *a s Cl.*	High level management. Low latency and Heterogeneity. Mobility using fog computing.	Edge computing enabling the IoT.	D	ODL, Onix and ONOS controllers SD Fog gateways SD-MEC OF-switches	Multiple identification and comm. technologies. Multiple SD fog GW for interoperability. Centralization security enhancement. Fine-grained flow services using FlowVisor or OpenVirtex.	Scalability. Infrastructure exposed to third party.
[111] *a m Cl.*	Low cost IoT. Enhanced scalability and interoperability.	SDN/NFV-enabled edge node for end to end SDN IoT services.	D	SDN controller IoT gateways OF-switches	ODL and OpenStack Nova/Havana service controller. GMPLS controlled optical network. Multi domain network architecture. Optimized packet response time.	Not a unified IoT platform.
[122] *m*	Efficiency and Scalability.	Dynamic manipulation of packets using NFs in docker.	D	SDN controller	NF-Lib for fast deployment of NFs. Improved scalability.	Third-party library functions may pose security risks.

*a: Architectural solution, m: Management solution, s: Security solution, D: Distributed, C: Centralized, Cl.: Cloud/edge/fog.

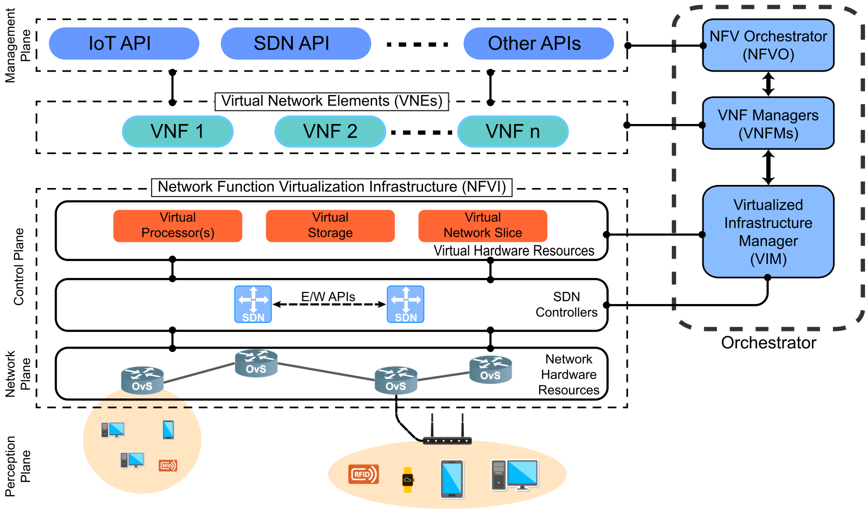


Fig. 3. A general SDN-IoT architecture with NFV.

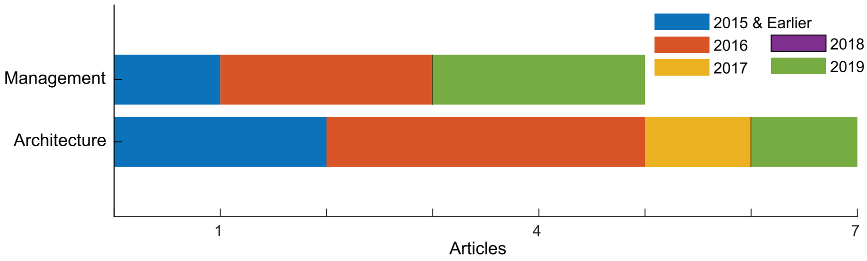


Fig. 4. Articles addressing the NFV-IoT solutions in past five years.

4.1 Architectural Solutions of NFV for IoT

Most of the architectural solutions are hybrid SDN-NFV designs, which take advantage of each other's capabilities but focus more on NFV techniques. These solutions are mostly focused on augmentation of function virtualization to (a) SDN based IoT systems or (b) cloud/edge systems. Table 6 gives the comparative analysis, where the architecture solutions are marked. It is important to note that some of these solutions also present security or management solutions, and hence they are highlighted in both subsections.

Li et al. [64] propose one such architecture following a top-down approach. It is divided into the application layer (e.g., services like Operation Support System/Business Support System), control layer (i.e., SDN controller with a distributed operating system) and infrastructure layer (i.e., IoT switches and gateways). The primary objective is to employ SD and NFV to meet IoT challenges, such as heterogeneity, scalability, security, and interoperability, by centralizing the control and virtualizing different IoT services. However, the proposal only discusses architectural details of how these services may be realized, and authors intend to study the organization and components of each part of the SDN/NFV-based IoT framework as a future direction.

Du et al. [32] focus on prototyping a context-aware forwarding/processing mechanism for IoT traffic management. This contextual information is distributed from both a sensor layer and an application layer to mitigate challenges of an IoT network related to scalability, discoverability, security, reliability, computational, and battery limitations. The objective is to enable multiple Mobile Virtual Network Operators (MVNOs) over shared wireless infrastructure. Hence, the architecture uses programmable switches to enable software-defined data plane services for MVNOs. IoT Gateway software ensures trailer slicing on FLARE [39, 75] platform, providing functionalities like IoT device discovery and connectivity, data collection and encapsulation, and context-aware packet forwarding/processing.

Mekikis et al. [72] propose an SDN/NFV-enabled solution to support industrial environments using a 5G tactile Internet. The solution supports heterogeneity for IIoT by softwarizing physical network function to VNF, which are dynamically created, updated, monitored, and deleted as per the network condition demands. The solution is a three-layered architecture that embeds intelligence at all layers. Test set-up implements SDN networking with local cloud support to coordinate VNFs at IIoT gateway. The field layer at the bottom consists of IIoT devices such as smart sensors using the 802.15.4 protocol. SDN switches are in the network layer to communicate with the compute nodes and IIoT gateways. Part of the field layer consists of a local cloud, which is an OpenStack-enabled ecosystem. It handles authentication, authorization, and accounting (AAA) and VM images for storage. However, the solution is limited to support the 802.15.4 protocol only. Performance evaluation shows significant improvements in handling latencies regarding services hosted at the IIoT gateway directly. Future work aims to extend the 5G radio access network to support OpenAirInterface project.

While considering the role of edge nodes, Ojo et al. [82] present an IoT framework based on virtualized elements in an SDN-enabled system. They utilize VNFs for several purposes, which are deployed on SDN/NFV edge nodes. By using these edge devices the framework can provide services such as rich user context (location information), low latency, high bandwidth guarantees, and rapid IoT device deployment. The MANO plane orchestrates control of the network infrastructure and the different network functions through respective managers. NFV can also be used to create virtual gateways, which will allow greater scalability, easier mobility management, and faster deployment. Although theoretically the models proposed in this work are sound, there is no implementation or evaluation available to realize the system.

Similarly, Vilalta et al. [111] propose an SDN-based NFV edge node model. The proposed edge node adopts an OpenFlow-enabled switch, controlled by the edge SDN controller. The OpenStack Nova handles the NFV framework through the Cloud/Fog Network orchestrator, which has two different orchestrators running under it: (i) the Cloud/Fog orchestrator, which deals with the edge cloud and metro controllers, and (ii) the Multi-domain SDN orchestrator, which deals with edge and data center SDN controllers. This entire orchestration consolidates NFV and SDN together to provide seamless network connectivity between deployed VMs to virtual switch at the edge node. The IoT gateway acts as a client requesting computing and storage services from such an edge node. The Multi-domain SDN orchestrator simulates OpenDayLight and OpenStack Nova to provide end-to-end network services. The work is limited to the edge nodes and data centers only; however, it can optimize the packet latency between the IoT Gateway and edge node.

Another similar approach toward edge networking is given by Salman et al. [96], which presents a fog computing architecture termed Software-Defined Mobile Edge Computing (SD-MEC) for integrating MEC with IoT, SDN, and NFV. SD-MEC is a four-layer architecture that includes an application layer, a control layer, a device layer, and a network layer. In this framework, the Software-Defined Function (SDF) gateway acts as an inter-operator between various protocols and network technologies. This simplifies the management process and enables heterogeneity abstraction, low latency, and mobility support. Applying the NFV features further facilitates management at the network level required in the MEC platforms. However, this work only gives conceptual information regarding fog architectures and for a specific use case scenario.

Insights: The study by Alenezi et al. [4] shows that the use of SDN and NFV significantly reduces the operational and energy costs of networks as compared to 4G networks. The works presented in this section are mainly architectures only, focusing on the scalability of IoT networks and reduction in processing/communication overhead also, as shown by the distribution in Table 7.

Table 7. Key Areas of NFV-based Architectural Solutions

Concentration Domains	**Literature** (D: Distributed, C: Centralized, Cl.: Cloud/edge/fog)
Middleware virtualization and Application programmability (for scalability)	[4] (C, Cl), [64] (C), [72] (D, Cl), [82] (Cl), [96] (D, Cl), [32] (C)
Resource Optimization	[4] (C, Cl), [111] (D, Cl)

Implementation and evaluation are two key elements missing from these solutions. Similarly, the coupling of SDN and synchronization of different virtual functions with orchestrator and control layers could lead to an improvement in the deployment of VNFs in IoT. It is important to note that the use of VNF has mostly been done in addition to SDN. Although this gives more programmability and control over the ecosystem, virtualization of IoT device functions in a local environment are an important research direction. Offloading of complex tasks as virtual functions to a gateway or other capable nodes can be very beneficial in some architectures, such as blockchain (consensus formation), image analysis, AR/VR applications, and so on.

4.2 Management of IoT Using NFV

Solutions that virtualize management functions for IoT networks are usually aggregated with SDN solutions, either as application processes or controller functions. In the following reviews, we discuss solutions that directly focus on the management aspect of an IoT network. In this regard, Balon et al. [8] propose a model for robust security and network performance management. They show a use case to build a private virtualized MVNO, which can easily be expanded and scaled for high-volume traffic and users. Their main effort is to provide a cost-benefit analysis of using MVNO. However, the article discusses architectural details and market analysis but does not give much information on the implementation of such MVNO services.

Batalle et al. [9] integrate NFV and SDN to reduce cost in IoT, where a centralized controller is responsible for routing that has a global view of the network. This work presents a novel design of a virtualized routing protocol using NFV infrastructure. It simply manages and reduces signaling overhead, particularly when inter-domain routing is required in an OpenFlow device network. It aims to reduce the number of connected and deployed devices, which will reduce the cost as well. The evaluation shows that it can reduce the number of flow entries by 50%, which improves scalability. However, robustness and performance may be affected. The experiments lead to several open research questions, the most important of which are routing algorithms for virtual hosts and routing policy optimizations.

Maksymyuk et al. [69] adopt an IoT-based network monitoring framework to manage the performance of 5G heterogeneous networks under different conditions. In this architecture, Radio Access Network functionalities are virtualized using NFV to simplify load balancing and spectrum allocation. Moreover, centralized intelligence of the SDN controller is used to implement interference-aware spectrum allocation. This allows better load balancing of smaller cells and manages user's mobility. This proposed framework has two main advantages. First, only relevant data will be subscribed by each network operator that can improve the existing monitoring system. It also supports multiple Mobile Network Operators. Second, the small size of the transmittable data block generates less traffic overhead.

Zhang et al. [122] propose an extension to OpenNetVM using Network Function (NF) management module that manages on-demand NFs in lightweight Docker containers. OpenNetVM supports flexible and high-performance NFV architecture for smart IoT platforms, enabling increased interoperability among NFs. The proposed NF management module uses service chaining for efficient and scalable packet processing. This may enable complex virtual services for deep analysis

Table 8. Key Areas of NFV-based Management Solutions

Concentration Domains	**Literature** (D: Distributed, C: Centralized, Cl.: Cloud/edge/fog)
Routing function management	[9] (C), [111] (D, Cl)
Service chain management	[7] (D, Cl), [122] (D), [4] (C, Cl), [111] (D, Cl)

within the network and may also remove the limitation of managing large volumes of IoT devices to some extent.

To meet the 5G future wireless network requirements, Atzori et al. [7] bring cloud computing services very close to the end-user by virtualizing the physical IoT devices to realize the Smart Device-as-a-Service (SDaaS) concept. It facilitates scalability, flexibility, and reusability of code and supports network traffic congestion control. NFV infrastructure renders OpenStack or Open Volcano APIs in the fog network, which drastically reduces network hops between server and client. This eventually preserves the security aspect as well. SDN controller configures the virtual switches in the edge/fog network. Due to reduced workload, the power consumption of IoT devices declines, eventually extending the lifetime of the battery. Since the proposed solution embraces many open source software projects, compatibility among them is very challenging.

Nguyen et al. [79] present a multiservice, multitenant, and multi-access solution to manage edge IoT devices by enabling VNF elements running on edge nodes. The virtual edge nodes can flexibly route traffic, aggregate, and split flows across any heterogeneous network. It offers efficient network operations and optimizes computing resource allocation for end-to-end IoT devices. SDN manages the backhaul bandwidth while the integration of cloud computing can further enhance the performance.

Insights: The virtual function can play a significant role in the management of IoT devices, networks, and associated services, especially in the key areas of service chaining and routing functionality. Most of the existing research has been focused on these points, as shown in Table 8. The concept of the device as a service, in reference [7] is very important and should be explored more, especially with the combination of the OpenStack platform. Other works, such as References [46, 102], use the virtualization of wireless interfaces, which is somewhat similar to the gateway abstractions in References [82, 95]. Several open research challenges require a more comprehensive and inclusive ecosystem. In the solutions discussed above, third-party services are utilized to manage and facilitate the network topology, which usually leads to compatibility issues. Open standards in this regard may be helpful. The research community may also work on developing SDN/NFV-based advanced real-time applications to manage and orchestrate IoT nodes, especially in the context of knowledge-based 5G (and beyond) mobile networks.

5 SOFTWARE-DEFINED INTERNET OF THINGS

In this article, we classify SDN-based IoT solutions and Software-Defined Internet of Things (SDIoT) solutions as two separate categories, with different architectures and scope. The difference among them is subtle but significant. SDIoT extends the Software-Defined (SD) approach to collect and aggregate data from network devices, sensors, and cloud platforms. The primary difference is that SDN-based systems only provide packet flow configuration for network devices, thus enhancing network connectivity. Hence, SDN-based IoT is limited to network layer virtualization. NFV implementation extends the network connectivity and security. The basic idea is to virtualize key NFs and place them on commodity servers. The next step is to connect them via a flexible SD infrastructure managed through a unified orchestration system. For optimization, service provisioning, scalability, performance enhancement, and rapid deployment, the whole IoT ecosystem

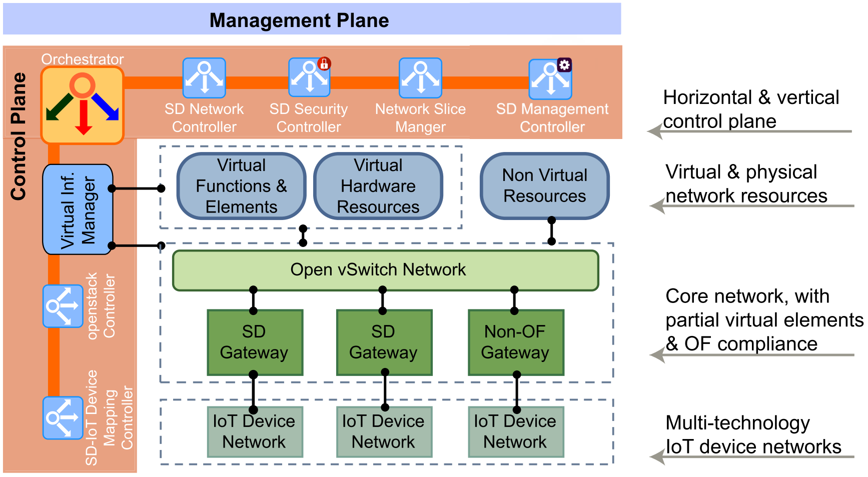


Fig. 5. A software-defined Internet of Things architecture, with horizontal/vertical control plane and integrated function virtualization.

can be virtualized in an SD paradigm. Finally, SDIoT solutions are not limited to a specific layer but range from device up to the application.

The generalized architecture of SDIoT is shown in Figure 5. Comparing it to Figure 1, it can be observed that the scope of virtualization has significantly increased, not only horizontally but also vertically. The control layer incorporates different domain-specific SD controllers, each executing specific tasks within SDIoT architecture. This reduces the burden on a single controller. In an IoT network, flow management is not the only challenge; thus, using dedicated controllers is more meaningful. Moreover, the control layer is extended vertically to add function virtualization at lower layers. Hence, the function virtualization orchestrator becomes an integrated part of the control plane. Protocols/APIs for SDIoT framework varies upon the nature of communication and the type of IoT devices connected to it. Some of these devices may even be virtual or have virtual functions running on them. A widely used OpenFlow protocol already exists to communicate between SD-controller and OF-switch, but it needs to extend its capabilities to communicate with IoT devices beyond the traditional OF switches. The application and management layer communicates with the connectivity layer through the NBI. This layer can also have a management-specific framework, which can enforce different policies through the programmable interface for SD controllers to execute. This framework can also enable different virtual functions at different layers of the SDIoT network for groups of different nodes.

The following sub-sections present different architectural and management solutions exploiting different SD controllers. Table 9 summarizes and categorizes SDIoT architecture and management-related literature, while Figure 6 shows the emphasis on software-defined IoT ecosystem in recent years.

5.1 Architecture Solutions

SDIoT architectural solutions use multiple controllers for providing different services. In addition to traffic flow management, these architectures focus on the data collection and analysis within the network. The tight coupling of such solutions with different types of controllers can be observed in the following.

To address the in-network data collection from IoT devices, Din et al. [30] propose an SDIoT architecture, which consists of data collection and management controller. The data pass through the Data Processing Layer, Data Management Layer, and Application Layer. The aggregated data, via

Table 9. Software Defined IoT Solutions and Their Classification

*Ref.	Objective(s)	Solution	*Control Plane Architecture	Benefit(s)	Limitation(s) / Future Work
[1] *a Cl.*	Enhanced packet fwd mechanism. Heterogeneous and Agile	SD-IoV solution.	D	Efficient packet forwarding. Effective for Smart City Application.	Flow cong. detection not considered. Vehicle positioning and trajectory environment missing.
[5] *s m*	Enhanced packet fwd strategy. Control for sensor nodes.	SD-WISE solution.	D *(SD-WISE controller)*	Virtualized SD-security elements. Overall resource usage optimization.	Specific to sensors.
[30] *a*	Data sensing, collection, and processing. Scalability and availability.	SDIoT architecture to analyze data of smart cities.	D	Hadoop ecosystem for load balancing. Data collection done using SDN and NDN.	Complex scheduling algos needed for cluster-based Hadoop systems.
[48] *a Cl.*	Reliability, scalability, security, and QoS.	SD-IIoT design to manage data exchange and delay.	D *(FloodLight)*	App.-specific approach for node performance and interoperability. Focus on network controllability	Optim. for more than 10 parallel connections not possible.
[51] *m Cl.*	Scalability, Heterogeneity, Agile, and Inexpensive.	SD solution for IoT to forward, store, and secure data.	D *(Multiple SDN controllers)*	Multiple SD application modules to facilitate IoT network.	Architectural design only. No implementation or evaluation.
[56] *m*	Sensing, security, and scalability.	A middleware solution for context-aware smart buildings using SD WSN.	C	Avoids single point of failure. Fast response to dynamic changes.	Prototype is limited to single building.
[60] *m Cl.*	Container-based solution for Open Stack and Kubernetes. Security, reusability, and Heterogeneity.	Chain services across SDN-enabled IoT network. Reduced maintenance overhead.	D	Supports dynamic service chaining to interact with container and VM domains.	Limited to Kubernetes and OpenStack only. Code is open source, but no performance evaluation.
[67] *a Cl.*	Sensing and robustness	SDIoT architecture for smart urban sensing.	D	Dynamic data optimization, processing, and transmission.	Application config. depends on shared sensor platform.
[76] *m Cl.*	Configuration, operation, and access control of cloud system.	Fleet management system using SDIoT cloud.	D	Overall resource usage optimization. Elastic policy-based configuration. Cost awareness.	Limited implementation and evaluation. Runtime SDIoT governance and resource usage in future.
[113] *a*	Reliability, standardization, and security.	SD-IIoT architecture for seamless data processing.	D	SD-data collection, transmission, and processing. Solution for illegal access and IoT mobility vulnerabilities.	Limited evaluation of the proposed solution.
[114] *m*	Scalability and reliability. Mobility.	Distributed overlay structure to support mobility and dynamic flow control.	D *(FloodLight)*	Mobility management, Handover optimization, and Distributed control.	Flow-scheduling optimization issues concerning backbone network.
[115] *a Cl.*	Scalability, Mobility, Openness.	Smart Home IoT device integration with SDN-based services.	C	Virtualization to simplify heterogeneity and complexity of diff. SDSH protocols.	Architectural design only. No implementation or evaluation.

*a: Architectural solution, m: Management solution, s: Security solution, D: Distributed, C: Centralized, Cl.: Cloud/edge/fog.

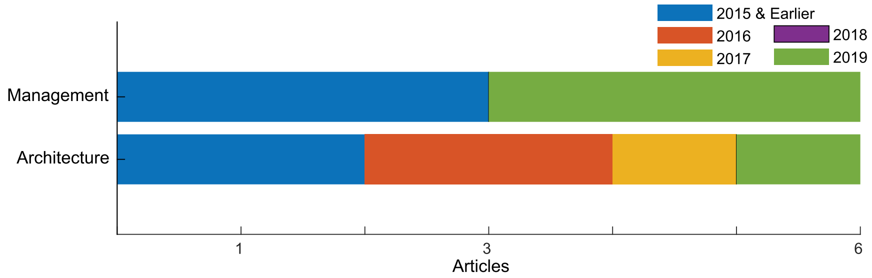


Fig. 6. Articles addressing the SD-IoT solutions in past five years.

various Aggregator Points (i.e., Zone, Local, and Global), are passed on to processing and management layers for real-time data processing and extraction. Since IoT devices generate large volumes of data, the proposed system utilizes a Hadoop Distributed File System for data storage and manipulation purposes. The work also uses Information-Centric Network [43] and Named Data Networks [53] to fulfill its requirements. The simulation results show some interesting aspects. HDFS gives high throughput and less processing time, even though they can still be improved using a cluster-based Hadoop system with efficient scheduling mechanisms. Extending the data collection to end devices, Liu et al. [67] propose an SDIoT architecture to separate smart urban sensing applications (for data collection) from the existing physical infrastructure, as most of the underlying network elements (sensors) are not SDN enabled. The authors divide the entire framework into three layers, i.e., physical infrastructure layer (sensors, smartphones, gateways, etc.), control layer (SD controllers), and application layer (IoT applications). SD controllers manage configuration for each hardware resource and provide the interface to standard API services for data manipulation. Each of these controllers can be replicated to enhance its robustness and can be physically placed anywhere for resource usage optimization. The specialized sensor controller has complete knowledge of underlying infrastructure and is capable of activating/deactivating sensors dynamically. The forwarding devices are OpenFlow enabled and programmable, and the SDN controllers are responsible for scheduling packet flow tables for forwarding devices and smart traffic steering. However, the cloud platform allows urban sensing data to be stored and processed. Cloud controller monitors and maps the underlying server resource pools. Although the architectural design is supported by case studies and qualitative investigations only, it shows promising possibilities to improve network resource utilization as well as dynamic data optimization, processing, and transmission.

By applying fundamental SD features like centralization, virtualization, and optimization, another similar approach is taken by Xu et al. [115]. They present an IoT-based software-defined smart home. It supports openness, virtualization, and centralization, with integrated heterogeneous devices. The controller acts as a management layer providing compatibility and API support. The architecture also uses virtualization technology to maintain uniform virtual abstraction of hardware resources of the devices in a smart home ecosystem. Moreover, it uses VNFs for access control mechanisms and load balancing. Although the overall architecture shows promising aspects, simulation or real-time experiments should be carried out to prove the effectiveness of the solution.

Hu et al. [48] propose a dynamic controllable solution for Software Defined Industrial IoT (SD-IIoT) with SDN features in it. The solution emphasizes the application-specific holistic performance approach of network nodes like field devices, gateways, and sensor clouds concerning connectivity and interoperability. Specialized QoS controller enforces QoS policies for the network backbone and field WSN. The network controller handles topology management and data updates. Also, the data synchronization controller and security controller are present. An additional Data

Table 10. Key Areas of SD-IoT Architectural Solutions

Concentration Domains	**Literature** (D: Distributed, C: Centralized, Cl.: Cloud/edge/fog)
Application virtualization	[48] (D, Cl), [115] (C, Cl)
SD-IoT data analysis in controller	[1] (D, Cl), [30] (D, Cl), [67] (D, Cl), [113] (D)

Manager module provides data management services, and the control module implements control plane functions. The authors show that latency can be reduced by 30% to 38% as compared to Amazon AWS. Similarly, Wan et al. [113] propose a Software-Defined Industrial Internet of Things (SD-IIoT) architecture utilizing SDN and industrial cloud. The physical layer consists of various kinds of hardware devices such as sensors, gateway, switch, router, and so on, while the control layer manages them. Only two controllers, SD controller and SDN controller, are used. The proposed SD-IIoT architecture also provides three major services of collection, processing, and transmission of data. Decision making is autonomous while data processing is software defined. As the system would deal with large-scale big data, the SD-IIoT service mechanisms require efficient data processing mechanisms/algorithms, which the authors aim to develop in the future.

In a different use case, Abbas et al. [1] propose the Software-Defined Internet of Vehicle (SD-IoV) with a novel protocol for V2X communication. The edge controller (through multiple RSU) is responsible for a set of road segments and vehicles on them. It then interacts with the SDN controller for path calculation in the whole network. The authors propose several new algorithms for different routing optimizations. It also use 4G/5G connectivity to forward packets. The evaluation does show some improvement in data collection and forwarding; however, it does not complete the application scenario.

Anadiotis et al. [5] introduce a Software-Defined Wireless Sensor networking (SD-WISE) approach, which is an SDN but can reach the sensor nodes beyond the virtual switch. Moreover, it adds function virtualization for different services of sensor nodes.

Insights: The contributions in this sub-section include IoT/IIoT concepts with SD features while focusing on the application level virtualization to support operations and data analytics, as depicted in Table 10. As the data generated by IoT devices have to be processed, thus this has become a key research area. Perhaps the most comprehensive solution in this area is given in Reference [67], which is very close to the generalized illustration of Figure 5. It is important to note that the details of most solutions mainly focus on incorporating APIs in the application layer to enforce decision rules on SD controllers and to exploit network virtualization features. Although this does increase the features of the solution, it can be viewed as positive and negative. On one hand, it takes away the control and intelligence from the control layer (making it more of an enforcer of rules), while, on the other hand, it allows a unified application for large-scale systems. Solutions such as in Reference [1] enable the use of 5G systems, but capitalizing on the programmability features of 5G design and coupling them with SD literature has not been done. Another important aspect is the vertical extension, as in most IoT networks the devices are not OpenFlow compliant, hence orchestrating them can only be achieved by it. Future work may also focus on the integration of VFs specialized for different controller types and their specialized placement in the topology. Moreover, the distribution of different controllers in the networks may improve performance and reduce communication latency with IoT devices. In this regard, inter controller communication may also require further improvement and standardization.

5.2 Management Solutions

Managing and configuring a diverse range of IoT devices can be a challenging task. To reap the benefits of network programmability and efficient resource utilization, a few works have focused on SD-IoT management solutions. The works discussed here focus on cloud resources, device clusters, and mobility management.

References [76] and [56] provide cloud and cluster management of an SD IoT network for specific scenarios. Nastic et al. [76] apply SD in IoT by creating abstractions wrapped in SD-APIs for different IoT devices. The proposed system directly interacts with the underlying physical IoT infrastructure (vehicular fleet). The objective is to have a unified view for configuration, access, and control of IoT cloud systems. The architecture presents fundamental building blocks for automating configuration and provisioning processes, which will eventually simplify IoT cloud operation management. However, the exchange of raw IoT data in the cloud needs a lot of computational resources and bandwidth. Future work may consider mechanisms and techniques to enable optimize resource usage of edge networking and allowing policy-based automation of security and data quality of SDIoT systems. Kathiravelu et al. [56] propose an architecture for Software Defined Building (SDB) [27], using smart clusters. This enables communication among IoT appliances within a multi-building campus to enhance their programmability and re-usability. It also uses the Software-Defined Sensor Network [68] to manage communication mechanisms between sensors and IoT appliances for system policy implementation. The addition of specialized IoT device SD controllers allows fast response to dynamic changes. Moreover, these controllers are distributed in a cluster that avoids overloading. Deployment in real-world scenarios is complex, and, hence, authors have left it for future work.

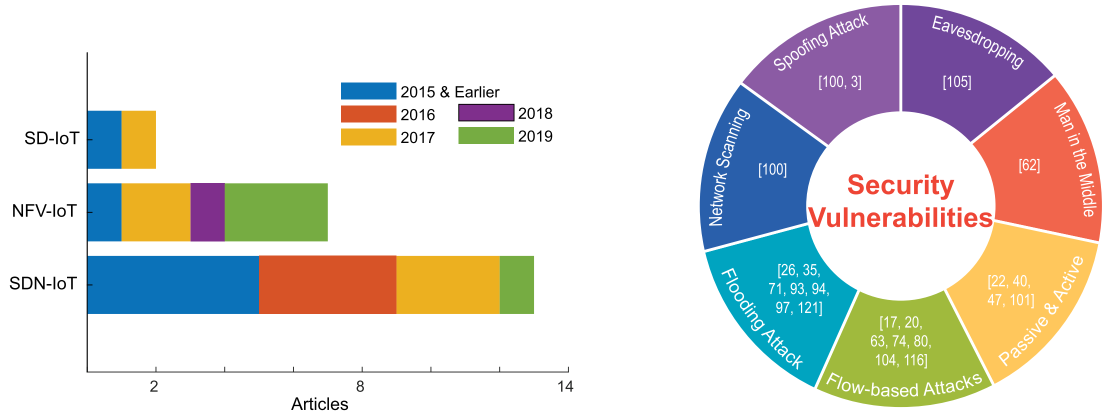
Another clustering approach is proposed in Reference [114] with a focus on mobility management. Instead of using completely centralized controllers in the IoT-based urban mobile networks, Wu *et al.* introduce a distributed overlay structure to support ubiquitous mobility management and dynamic flow control where the entire SDIoT network topology is divided into different geographic chunks or clusters. The authors focus on the logical centralization of controllers while they are physically placed at different locations. An orchestration controller is used to communicate with local controllers. The mobility of sensor platforms is also managed through the orchestration controller. However, for the backbone network, further provisioning of flow-scheduling optimization is not considered.

For a more comprehensive solution and to address the needs of heterogeneous nature of IoT applications and objects, Jararweh et al. [51] propose an SDIoT framework with an enhanced IoT management layer. This model enhances several important aspects like security, storage, and traffic forwarding. It has three main components. First, the physical layer deals with all physical devices like sensors, servers, switches/routers, and security hardware. Second, the control layer is the core of the proposed prototype to manage and coordinate among different SD controllers, i.e., IoT controllers, SDN controllers, SDStore controllers, and SDSec controllers, to abstract the management and control operations from the underlying physical infrastructure. Third, the application layer through NBIs combines fine-grained user applications to facilitate access control and data storage mechanisms. Additional controllers can be added to tackle sophisticated load balancing and inconsistency issues and to deliver fast response time for many requests within the network. The authors in this prototype conceptually use SDN, SDStore, and SDSec to build the architecture only.

Kouchaksaraei et al. [59, 60] introduce a multi-domain orchestrating framework that can manage OpenStack [61] and K8 [84] infrastructure, which can collaboratively manage container-based and VM-based VNFs. The objective of the proposed solution is to chain services across SDN-enabled networks. It eventually works as an inter-domain orchestrating tool to manage all services across the multi-domain IoT network. The solution improves the reusability of technologies and reduces maintenance overhead. But the solution is only limited to comply with K8 and OpenStack infrastructure.

Table 11. Key Areas of SD-IoT Management Solutions

Concentration Domains	**Literature** (D: Distributed, C: Centralized, Cl.: Cloud/edge/fog)
SD-IoT middleware	[56] (C)
Virtual service chaining	[60] (D, Cl)
SD-IoT data management	[5] (D), [51] (D, Cl), [114] (D)
SD-IoT cloud management	[76] (D, Cl)



(a) (b)

Fig. 7. IoT network security solutions: (a) Approaches for different solutions in past five years. (b) Vulnerabilities addressed by the solutions.

Another OpenStack solution by Chen et al. [23] presents a cloud-based OpenStack platform that integrates Kubernetes-enabled NFV and MANO orchestration to manage and scale IoT/M2M communications. Kubernetes manages the scaling functionalities through the containerization mechanism, while NFVO integrated with OpenStack Tacker configures VNFs for resource allocation. The scalability of IoT and M2M communication can significantly increase in this way.

Insights: From the solutions discussed above, it is evident that most of the work is focused on managing a single element of the network, and most research focused on data management as shown in Table 11. Reference [51] presents a more comprehensive solution but does not give implementation details. Most of the other research contributions primarily focus on extending APIs in the application layer to enforce decision rules on SD controllers and SD-Gateways and to exploit network virtualization features. However, in the presence of multi-vendor solutions at application, control, and data layers, standardization for communication interfaces (NBI) becomes very critical. To date, there has been no effort to do that, and most solutions only use REST. Moreover, functionalities specific to IoT devices should also be part of overall management architecture such as mobility and resource management.

6 SECURITY SOLUTIONS

Reference [36] presents a comprehensive survey and classification on SDN and NFV issues of IoT networks. Hence in this work, we present only those solutions that are extremely relevant to the classification of this work. Based on the earlier classification, we have grouped the solutions into SDN-based, NFV-based, and SD architectures. Each subsection has an accompanying table, which shows the limitations and comparative analysis, while Figure 7(a) shows the distribution of solutions in the past five years. As SD-IoT is relatively new, and hence it has not received much attention from the research community. Figure 7(b) shows the vulnerabilities addressed and the specific literature that proposes solutions for them.

6.1 SDN-based Solutions for IoT Networks

The traditional security mechanisms for external threats are deployed at the network edge, such as firewalls. However, the dynamic changes in network topology as a result of IoT nodes joining-in and moving-out, require them to be repeatedly re-configured and updated. Similarly, new internal threats from rogue devices or vulnerabilities in software or hardware of devices require continuous monitoring and patching [119]. Hence, the security parameters for both internal and external threats may need to be reconsidered with the flow of technological advancement. The following literature discusses different proposed solutions for SDN-based IoT security issues. Table 12 shows comparisons among them. We group these works into different categories: architecture and protocol-related vulnerabilities, flow-based security issues, Usecase-specific security issues, and other miscellaneous attacks and vulnerabilities that expose the network.

Architecture and Protocol Vulnerabilities: In an SDN environment, the communication between IoT-based devices and servers can be blocked by new flow attacks that contain a significant amount of unmatched packets injected into the routing system. This leads to the processing of an excessive amount of data packets in both the control plane (by overloading the controller) and data plane (by overloading resources at OF switches). To solve this issue, Xu et al. [116] presents a security framework to defend against such suspicious flow attack for IoT networks. The controller acts as a security middleware to filter new-flow vulnerabilities, such as DDoS [10], switch to controller communication flooding, and flow table flooding. However, large-scale deployment can be a challenge for it. Another similar approach is for OpenFlow-enabled systems presented by Shin et al. [100], which allows rapid design and deployment of modules to detect and mitigate threats.

Sandor et al. [97] presents an IoT-based hybrid network framework along with a redundant path switching algorithm using SDN's adjustable routing feature, which would protect against DoS attacks. The architecture is hybrid, because it includes SDN switches and non-SDN topology segments that contain both types of Entry Point (EPs) and communication edges. Similarly, Flauzac et al. [40] proposes a solution that is mainly designed to enhance the security of SDN controllers and to solve the scalability issues in multiple IoT-based domains. The work combines wired and wireless networks and further extends its solution to an ad hoc enabled network and IoT devices like sensors, smartphones, tablets, and so on.

In a distributed network scenario, Gonzalez et al. [42] introduces a proposal that is adequate for an IoT cluster environment by establishing groups of sensor nodes. Instead of using a traditional approach of the static firewall to block a possible attack, the authors presented an SDN-based routing protocol and a dynamic firewall termed as Distributed Smart Firewall that can apply the functionality of an SDN controller. Another SDN Controller clustering approach by Shuhaimi et al. [101] deals with challenges like availability, heterogeneity, security, and privacy in IoT. It also proposes a multi-step novel algorithm, to select SDN Cluster-Head (SDNCH) that works as an SDN controller. Its job is not only to manage and control network traffic but also monitors and prevent attacks from inside and outside domains by securing the whole SDNCH domain.

Network access control is a security mechanism that limits access to authorized devices only. Hesham et al. [47], using the SDN controller, presents a novel network access control service for IoT sensor networks and M2M communication. The solution gives a predetermined network access policy for each device, implemented through the controller for authentication and authorization.

Flow-based Security: Dataflow-related challenges of IoT devices and systems have been described by Bull et al. [20], where SDN gateways are used in a distributed structure to monitor data traffic and flow characteristics. The authors propose a method to identify and reduce anomalous

Table 12. Comparison of Security Solutions Using SDN for IoT Networks

*Ref.	Objectives	Vulnerability	SDN Controller	Switch Type	Implementation and Evaluation Details	Operational Layer(s)
[20]	Anomalous packet flow detection.	TCP flooding, DoS, ICMP attacks on IoT device.	POX	OF 1.3 Switch	Flow monitoring and periodic checking to counter TCP and ICMP attacks. Mininet-based emulation.	Datalink Network
[22]	Secure meta-data and payload within layers. Privacy and Confidentiality Integrity and Authentication	Packet injection. Eavesdropping.	Centralized controller	OpenvSwitch	Novel encryption for payload. Mitigates several active and passive attacks. SDN for routing over multiple topologies. Node sleep and sync. mechanisms.	Datalink Network.
[40]	Distributed routing. Distributed security rules.	General security issues.	Distributed controller	OF Switch	Multi-domain access control SDN architecture. Provisioning security for IoT objects.	Datalink Network
[47]	Novel network access control mechanism.	Unauthorized access to network devices.	ODL	Pica8 Switch	Testbed with in-band topology (merged control and data plane) to enable connection between clients and authentication service.	Datalink
[62] *Cl.*	Detect Man in the Middle attacks	TLS vulnerabilities	Floodlight	OF 1.3 Switch	Bloom filters-based SDN and extended OF to detect MitM using Mininet.	Datalink.
[63]	Detection of anomalous behavior in packet flows.	Neptune attack.	SDN controller	OpenvSwitch	Flow monitoring, periodic checking, and flow installation mechanisms. Regression model with fused lasso.	Datalink Network
[80]	Identify and block attacks.	Unauthorized access of smart home devices.	Floodlight	OF Switch	Identify suspicious flows to Smart Home IoT devices.	Datalink
[97]	Dynamic switching among redundant entry points.	DoS attack	Floodlight	OpenvSwitch	SDN-enabled hybrid net. with auto-switching and routing.	Network
[100]	Dynamically changes path against malicious scans.	Flow-based attacks.	NOX	OF-Switch	Detect and mitigate malicious flows with min. overhead.	Datalink Network
[101]	Reduced hardware usage. Enhanced security and privacy.	Third-party applications Untrusted data, Privacy	SDN controller	OF Switch	IoT and SDN integrated algorithmic model against inside and outside attacks.	Datalink
[104] *Cl.*	Network level monitoring to detect flow-based anonymous packets.	Two new Python-based emulated attacks.	SDN controller	TP-Link SDN-enabled gateway	Experimental testbed using C programming.	Datalink Network
[105] *Cl.*	Device Monitoring and Control.	Eavesdropping Remote access Privacy, MitM	Floodlight	OpenvSwitch	Prevention of eavesdropping and packet injection.	Network
[116]	Detection, Mitigation.	Suspicious flow attack.	ODL	OpenvSwitch	Testbed of IoT centric OF switches with ODL. Novel packet filtering algo in Matlab.	Datalink.

*Cl.: Cloud/edge/fog.

behavior and enhance QoS by the SDN-based IoT gateways. Sivanathan et al. [104] elaborates the differences between flow-based monitoring approaches and packet-based approaches to prevent vulnerabilities in smart-home IoT devices. Based on the flow-level characterization of IoT traffic, the authors present a system containing SDN-enabled gateway with a cloud-based controller to identify malicious IoT activity in the home networks. Li et al. [63] present an intrusion detection system to predict anomalies that are determined from traffic flows, for better access control. Based on a given input the model observes the deviation from normal and the policy generator installs the rules for such flows. The solution is unique as it uses regression models to determine abnormalities.

Use-case-specific Security: The usage of SDN in IoT for application-specific use cases is very important. Sivaraman et al. [105] illustrates that a significant number of IoT-based home appliances such as motion/monitoring sensors, smoke alarms, and smart lights, lack basic security functions. The authors argue that security implementation needs to consider various kinds of factors like device capabilities, mode of operation, and manufacturer. They propose a prototype, Security Management Provider (SMP), that can control access to data on devices by applying dynamic or fixed content-based policies to identify attacks (e.g., eavesdropping, spoofing, etc.) at the network level. Nobakht et al. [80] proposes an Intrusion Detection and Mitigation (IoT-IDM) solution, providing network-level prevention mechanism against malicious or suspicious ad hoc objects from the external network domains to access Smart Home environment. They propose the use of machine learning to detect attacks from learned signatures. Although it can be highly effective, is generates more traffic and is limited to selective smart devices. There can be other security issues, which stem from application patches, specific target users, or data-specific vulnerabilities. However, SDN-based solutions may not be the optimal choice to address them, as SDN-based solutions are more traffic oriented.

Miscellaneous Security Challenges: Chakrabarty et al. [22] proposes Black SDN to secure SDN-based IoT networks. The Black SDN approach encrypts both payload and packet header at the network layer in a single controller environment. It also helps in communicating with different resource-constrained IoT devices through Black packets. This method can mitigate several passive attacks like inference and traffic analysis attacks and also secures metadata that correlates with each packet or frame of an IoT end-to-end device communication, hence improving payload efficiency. Although Black SDN demonstrates better security as compared to traditional SDN, an increase in traffic between IoT devices and controllers may complicate traffic engineering.

In addition to this, there are several other SDN-based solutions in the literature that aim at mitigating specific attacks. To detect man in the middle (MitM) attacks in Software-defined IoT-Fog networks, Li et al. [62] proposes a lightweight solution by modifying the existing OpenFlow protocol. This solution mitigates, (i)flow redirection in the data plane, (ii) information collection from the data plane, and (iii) disrupting network view of the controller.

Insights: In this section, several security issues and solutions have been discussed, which primarily rely on the SDN controller to enforce policies. These prevention mechanisms are mostly developed as an external module to cooperate with the SDN controllers. The research community may focus on possibilities to integrate these modules inside the SDN controllers to achieve enhanced scalability. Efforts may be taken to focus on more real-time evaluation against different threat vectors, which can help determine the status and effectiveness of solutions. It is important to note that some of these solutions, such as in References [63, 80], use regression techniques and machine learning, which can be highly efficient in mitigating a diverse range of attacks. However, the basic limiting factor is the single point of control. There are three main security enablers, which can be addressed through SDN-based solution in IoT:

- *Firewall and Intrusion Detection*: Traditional IoT implement them at the edge, and the same technique is employed in SDN-based IoT systems. Anomaly and threat detection are entirely flow based, which also requires random flow monitoring.
- *Packet Analysis*: Controller is responsible for it or may delegate to a management layer application in the cloud. However, this increases traffic flow and processing time at the controller.
- *Authentication and Authorization*: Traditionally done through specialized Certificate Authority, but in the SDN-based system, it can be done through controllers. As controllers are designed for flow management, this becomes an added (and difficult) responsibility.

6.2 NFV-based Security Solutions for IoT

Security solutions for IoT have garnered significant research attention; however, the use of NFV to leverage virtualization has seen very few solutions. Primarily, the use of virtual functions is to extract the security modules from fixed hardware/software locations and implement them in the cloud. From there they can be easily redeployed, scaled, and optimized for different applications. Reference [36] classifies them into solutions for decoupling from hardware, scalability, mobility, and service chaining. Here we discuss solutions that are network-oriented, summary of which is given in Table 13.

Massonet et al. [71] propose an extended federated cloud networking architecture for edge networks and connected IoT device security. The security solution utilizes lightweight virtual functions and Service Function Chaining (SFC). The IoT gateways in the edge networks are responsible for implementing global security policy by creating a chain of VFs for different purposes, such as firewall and intrusion detection. To secure the IoT-Cloud network slices, a module is implemented inside the IoT network controller. Al-Shaboti et al. [3] proposes an ARP server within an SDN-based architecture to enforce access control for smart home IoT networks. The ARP Server is a virtualized trust entity implemented as a virtual function for defense against ARP spoofing attack and network scanning.

Sairam et al. [93] use virtual functions to implement machine learning algorithms for securing different IoT devices in a home network. The deployment of virtual functions is done on edge gateways, where IoT devices forward their data. Zarca et al. [121] proposes a secure architecture with NFV in smart buildings that adopts policy-based cyber-security framework, capable of resisting both active and passive attacks like replay/masquerading attacks, tampering attack, malware injection, Zero-day vulnerabilities, man in the middle attack, distributed DoS attacks, sniffing/eavesdropping via AAA system, and analyzing the logs. Farris et al. [35] presents an SDN/NFV-based approach for service monitoring and management and on-demand policy implementation through the SDN controller. The solution enables dynamic reconfiguration and adoption of policy implementation in case of suspicious vulnerabilities traced.

Boudi et al. [17] present a unique lightweight virtualization solution provisioning security-as-a-service at the edge network for constrained nodes. Virtualization is implemented through a containerization mechanism to deploy virtual security functions on top of AP or IoT gateways. Montero et al. [74] also provide an edge network multi-tenant user-centric approach for security profiles. End-users can define their policies from a remote location, and the system performs a translation of policies for anomaly analysis and reconciliation.

Insights: NFV or SDN domains have different elements, applications, orchestration managers, virtual functions, communication APIs, and so on. A malicious or compromised element in any of them could have serious consequences for the whole system. As an example, a malicious VNF by a compromised software vendor, a compromised hypervisor, or MANO component could potentially compromise the entire IoT network. If these elements are well secured, then integrity,

Table 13. NFV-based Security Solutions for IoT and Their Classification

*Ref.	Objective(s)	Solution(s)	*Control Plane Arch.	Controller and Switch	Implementation, Evaluation, Benefits	Limitation(s)
[3] *s*	Enhanced security and latency.	IPv4 NFV-based ARP server for preventing ARP spoofing and network scanning.	C	Ryu controller	NFV dispatcher for packet inspection. Secure NFV-based ARP operation. Host and port mapping independence. WiFi and Ethernet simultaneous use. Reduced packet processing delay.	Only ARP attacks. IPv6 for IoT not considered.
[17] *s Cl.*	Enhance security.	Container-based security services for AP/IoT gateways on edge.	D	—	Implem. focus on packet processing and network utilization. Sig. reduction in CPU and RAM usage for security services.	Edge computing only.
[35] *s Cl.*	Improve security and reliability.	SDN/NFV for IoT system protection.	D	SDN controller	Dynamic reconfiguration, adoption, and policy implementation. Dynamic security VNF deployment.	Architecture design only.
[71] *s Cl.*	Enhance security.	NFV/SFC approach.	D	SDN controller	Integrated agent in IoT network controller and gateway. Security VNF within the federated IoT-cloud.	Architecture design only. Evaluation left as future work.
[74] *s Cl.*	User centric security. Different policies.	Multi-tenant user-centric VPNs.	C	—	Unique translation method, anomaly analysis, and reconciliation. Privacy policies and profiles synced on multiple devices.	Limited in multi-domain scenarios and user mobility.
[93] *s Cl.*	Reduce threats and resource usage.	NFV-based edge traffic security solution for IoT.	D	SDN controller	VNF improves security. ML with cloud deployment.	Very few types of attacks addressed.
[121] *s m Cl.*	Monitor, detect, and mitigate threats.	SDN/NFV solution to safeguard IoT devices.	D	SDN controller	Policy-based security measures. Reduced delays and latency within the IoT networks.	Future work on reactive VNFs for attack mitigation.

*s: Security solution, D: Distributed, C: Centralized, Cl.: Cloud/edge/fog.

confidentiality, availability, access control, and accountability can be well preserved. Research work on access control and packet inspection mechanisms, needs further investigation, especially for resource-constrained IoT devices, and NFV can effectively be used for them. The following are main security enablers that can be addressed through NFV solutions in IoT:

- *Firewall and Intrusion Detection*: Rather than implementation at edge or controller, virtual functions can modularize the different components (i.e., flow filtering, signature detection, etc.) and implement them as software abstractions. These can be highly scalable.
- *Packet Analysis*: Virtual packet analyzers can be customized for different protocols and applications. Furthermore, the use of machine learning algorithms in the cloud can significantly increase the analysis output.
- *Authentication and Authorization*: AAA can become more flexible as compared to traditional IoT or SDN-controller-based solutions. This can be considered as virtual trust authorities that can be customized for different domains and applications.

Table 14. Software-defined Security Solutions for IoT and Their Classification

*Ref.	Objective(s)	Solution	*Control Plane Arch.	Benefit(s)	Limitation(s) / Future Work
[26] *s*	Enhanced security and reduced cost of security operations.	SD-security solution.	C	Virtualized SD-security elements. Context-aware security. Security component configuration.	Traffic overhead optimization.
[94] *s Cl.*	Security, Privacy, and Connectivity.	Security sol. for SDIoT using SDN and NFV.	D	Slicing techniques. Cloud/edge computing. Low latency, high throughput and scalability with location awareness.	Inter access-controller connectivity.

*s: Security solution, D: Distributed, C: Centralized, Cl.: Cloud/edge/fog.

6.3 Software-defined Security for IoT

Enforcing policies for security and access control in large-scale networks can be made easier by programmable interfaces. An efficient solution can be defined by adding a dedicated security controller in the SD infrastructure for the IoT network. The following solutions, as summarized in Table 14, focus on a similar concept.

Salman et al. [94] discusses security and privacy requirements of IoT and proposes a framework for security that uses (a) software-defined and virtual function technologies for virtualization and (b) slicing techniques for isolation of the network. Their architecture consists of six layers and two types of controllers. The core controller acts as a global network OS, while the access controller provides a dynamic control model to support IoT device communication. However, inter-controller connectivity and seamless integration of modules can be enhanced.

Similarly, Darabseh et al. [26] addresses the challenges of providing multiple levels of protection and efficiency. They propose a centralized Software-Defined System (SDSys), which groups different SD Security (SDSec) systems. These SDSec are software abstractions of security mechanisms embedded in hardware devices in the data plane. The system is software defined, because the SDSec_Controller (i.e., software-based POX controller) can manage diverse data plane resources regardless of their vendors. Extension of SDSec deployment by using specialized distributed controllers can reduce the overloading of a single point.

Insights: The solution presented here mainly focused on preventing DDoS attacks, access and congestion control mechanisms, and slicing techniques. Future work may include developing APIs that may interact simultaneously through the NBI, SBI, and E/WBI to enforce security decisions and rules to the SD controllers/gateways. It may also be able to exploit network virtualization features to assign VNFs for preventing different threat vectors. The solutions should be able to defend against a wide range of threats, providing a global view of IoT nodes beyond virtual switches. The following are main security enablers that can be addressed through NFV solutions in IoT:

- *Firewall and Intrusion Detection*: Rather than implementation at edge, controller, or as a single virtual function, there can be a dedicated firewall controller with an intelligent anomaly detection mechanism. It can then coordinate with the orchestrator to deploy virtual functions at appropriate network locations.
- *Packet Analysis*: Packet analyzers can be pushed down the vertical extension to check for anomalies at the originating networks. This will significantly reduce unnecessary traffic redirection for deep packet inspection in the cloud.

- *Authentication and Authorization*: Similarly, authentication and authorization controllers can be dedicated to determining the appropriate access rather than forcing a single controller to implement all solutions.

6.4 Blockchain and Network Virtualization

In recent years, the use of blockchain (BC) in different fields of computing has gained significant attention, and IoT is no exception [13, 14]. Several works combine SDN and NFV with blockchain for IoT applications (for security and otherwise). Here we analyze this literature to see how it can be leveraged for IoT applications. It is important to understand that the blockchain is an entirely different technology, as compared to SDN and NFV, and hence using it to achieve security benefits is not straightforward.

References [52, 98] combine SDN-based or NFV-based IoT networks with blockchain, but there is no interaction among them. Blockchain works at the application level in securing the transaction among IoT devices, while the SDN or NFV only facilitates data networking.

The following solutions take this one step further, where the elements of the BC network are implemented at different components of a software-defined network for the IoT. It is important to note that such solutions are not limited to IoT only, and other networks can also benefit from them. Pourvahab et al. [87] present an SDN-based solution, which provides forensic auditing of packets at controllers where blockchain is used. The controllers are part of the peer network. Kataoka et al. [55] use BC to obtain a list of trusted services at the controller but do not elaborate on the working of blockchain for this mechanism or how the BC and controller interact. Similar work is done by Qiu et al. [90] in a software-defined industrial IoT environment, where the controller requests the BC to form a consensus among different controllers for synchronization of data.

Rebello et al. [91] presents the concept of virtualizing an IoT Blockchain itself as an orchestrated function chain. Although the blockchain is not completely virtualized, the initial results show a scalable solution.

Insights: Blockchain is a relatively immature field, although the number of publications for its applications is now several thousand. To reap the real benefits in IoT along with the use of programmability, it should be looked at from two main aspects: (1) Programmable blockchain elements such as peers, miners, ordering services, consensus algorithms, smart contracts, and so on. By making re-deploy-able virtual functions and adding a peer layer (similar to the control layer) with APIs for element interaction, a true programmable BC can be achieved. (2) The use case of IoT determines the BC solution. For example, a BC for e-healthcare would be quite different from that of industrial automation. In this regard, it is important to get out of the crypto-currency bubble and look at it as a distributed ledger technology. Solutions for providing blockchain-as-a-service would be interesting, given that the service is highly programmable and modules for consensus and contracts can be easily replaced.

7 LESSONS AND FUTURE RESEARCH DIRECTIONS

The fundamental objective of this article is to collect, categorize, and analyze different software-defined and virtual function solutions for the Internet of Things. From this analysis, we have identified some key lessons and possible research directions. The key lessons learn can be grouped as follows.

Software defined: SDN and NFV alone cannot fulfill the programmability requirements of modern networks. They have to be combined. Most of the solutions capitalize on a hybrid design but restrict themselves to solving specific problems. The way forward is to have a software-defined IoT, which not only manages flows but also enables other virtual services of the network. Moreover, this software definition has to be use case specific.

Gateway and Interface Virtualization: Most IoT networks connect to the Internet through (single or multiple) gateway devices. By virtualizing them (similar to a controller) significant improvement in the performance of multiple parameters can be achieved. Moreover, the possibility of virtualizing interfaces only can be significantly beneficial in multi-technology networks. Their replication and placement is an equally important element in software-defined IoT systems.

Vertical Extension and Inter-element Communication: Perhaps the most important lesson is to extend the orchestration vertically so that devices in the perception plane can also be managed. This will make the gateway and interface virtualization easier and increase the granularity of the control. Moreover, communication among these diverse elements has to be efficient.

Based upon these lessons and the earlier analysis, the success of SDIoT depends on best practices and requires improvement in different layers of the overall system, and hence we classify the future directions based on layer and components.

7.1 Application Layer

This is the topmost layer and is mainly responsible for user/administrator interaction and other generic application models for enforcing and configuring different policies in lower structure. Some of the core research directions are as follows:

- In the past, SDN applications have been written specifically for certain functionalities and only for specific controllers. This creates a major bottleneck, as there is no standardized application development framework available. Such a framework will be highly beneficial for both the research community to build test applications and also for industry in the rapid deployment of SD-IoT networks.
- Most existing controllers use REST API for communication with the application layer. Unlike SBI there has been little to no effort in standardizing NBI. This effort will certainly be an important step toward widespread adoption and development. An important research element in this regard is to allow diversified application and controller capabilities. As applications and controllers are both specific to different functionalities in the SD-IoT framework, the standardized NBI must be flexible enough to accommodate different types of communications.
- Most of the focus with regards to security has been on securing flows and attacks on networks. Hence, security policy enforcement has been extensively studied. However, the security of application modules is also as important as prior. A compromised application module can misconfigure and severely compromise the whole SD-IoT ecosystem.

7.2 Control Layer

This is the main focus area of SD-IoT and will require major research and contribution efforts. It is not possible to list all potential directions, and, hence, we list the major concerns for the control layer in SD-IoT.

- Communication among different elements of the control layer is an important aspect. In a traditional software-defined network, each controller independently controls a domain. Controllers in hierarchy usually use proprietary methods of inter-controller communication. However, in SD-IoT there are multiple types of controllers for the same domain. Besides, the control layer may use multi-vendor solutions, and hence a standardized interface is an important research direction. Communication with other domain controllers may also be investigated for efficient and optimized communication. Reference [118] did some interesting work in this regard that can be a good starting point, and Reference [11] extends OpenStack for edge networking.

- In traditional SDN, a single point of failure of the control layer is avoided by using the back-up controller(s). However, due to diversity in SD-IoT controllers, having a back-up controller of each controller may become costly, hence requiring investigation of cost-effective redundancy solutions. Moreover, architectures that are scalable and less complex would be required.
- SBI is a major element of the control layer. OF has been a de facto interface for the network controller to data plane communication. In light of diverse SD controllers, the suitability of OF may need reevaluation. SBI, which can effectively work for all types of controllers and devices, will be another interesting direction. At the same time, the SBI should be able to reach IoT devices beyond virtual switches. OF does not connect hosts but only allows flow installation on switches. In an IoT network, the mobile devices and AP may also need configuration and other policy enforcement. This requires enhancements to OF or needs new SBIs that can reach and configure the end devices.
- Efficient use of network function virtualization is also a key research direction. Function chaining for various controller processes may enhance the performance and allow better control in the network. As the vertical control layer implements VNFs, their orchestration with the horizontal controllers is also an open research challenge.
- In addition to other control layer challenges, the security of control layers itself, its elements, and communication is extremely important. The security controllers should not only focus on the security of data plane and network devices but should also ensure logical element security. Research in this direction will have a major impact on SD-IoT networks.

7.3 Controller Perspective

SD-IoT will consist of several controllers designed for specific operations. This will open several research directions to be explored.

- Placement of a controller or other control layer elements is a less researched area, mainly due to a single network controller. In SD-IoT networks, the number of controllers and topological structure of IoT devices may require a closer look at the placement in topology for different controllers.
- IoT networks will comprise of hundreds of devices (if not thousands) in a single SD domain. Hence, the scalability of controllers is an important factor. This will include not only scalable architectures but also programming languages, capabilities, storage, and processing at controllers. As there are multiple types of controllers; hence, scalability and coupling at a large scale will be a very interesting research direction.
- Synchronization of controllers and their policies will also be an interesting challenge. Furthermore, it will be equally interesting to evaluate the requirements of the domain and then utilize only those types of SD controllers that are required. Vertical versus horizontal deployment of controllers and associated VNFs may also present interesting design options.
- Controller virtualization is an important element in software-defined IoT systems. Virtualizing multiple controllers and coordination among them is a challenging task. Similarly, the placement of virtualized elements in core or edge networks will be an interesting research issue.

7.4 Management Perspective

The nature and properties of IoT networks have highlighted some newer research challenges, which were not evident in traditional SDNs. In a complete SD-IoT system, these will require significant attention from the research community.

- *Mobility:* In a single SD-IoT domain there may be multiple edge networks, with dozens of diverse mobile IoT devices with high mobility and limited resources. Some solutions have tried to address mobility in SDNs; however, in a hybrid ad-hoc/infrastructure environment with different physical layer technologies, it will present new research dimensions. Efficient and quick topology discovery in the mobile domain, path configuration, hand-over, and other scalability challenges should be further investigated.
- *Device configuration:* The edge and access network in an SD-IoT network will comprise heterogeneous mobile devices. A major challenge is to configure them according to policy dictated by the application layer. This also requires significant research before a unified framework can be developed.
- *Virtual functions:* Virtualization of different network functions will be an integral part of the SD-IoT ecosystem. Hence, their management in control plane, distribution, virtualization, and integration with other layers and APIs is a major research area.

7.5 Technology Interaction and Complexity

Most of the previous challenges and directions also deal with the complexity of overall architecture, but the research community needs to look at the integration of other technologies in the overall ecosystem, such as fog/edge computing, cloud computing, crowdsensing, blockchain, and so on.

- Crowdsourcing techniques can benefit extensively from SDIoT networks. The functions for task advertisement, auction, bidding, and offloading can be easily implemented through virtual functions and orchestrated by a crowdsourcing controller placed at the edge node. Such an architecture can enable the rapid deployment of sourcing tasks and the collection of data. However, this will certainly require further research in the specific controller design, virtualization of such controllers, and security among other challenges. This will also increase the complexity of the overall SDIoT framework, hence requiring more scalable systems.
- The diversity in devices and physical layer technologies, including the specialized IoT networks such as vehicular networks, must be considered in the overall design. Similarly, the use of content-centric (non-IP) communication would require controllers and interfaces to be re-designed. The use of programming languages such as P4 may be beneficial as compared to OpenFlow interfaces only.
- Blockchain is a relatively new area for IoT but may prove to be extremely beneficial in financial transactions and other private blockchain trades. Potential research directions may involve virtualization of complete peers/mines, offloading of complex mathematical functions, and proof of work to other nodes via virtual functions, virtualization of blockchain ledger, and so on. SDIoT may also pave the way for hybrid blockchains for the Internet of Things.

8 CONCLUSION

Software-defined networks have seen extensive deployment in data centers and core networks, where they have been mostly used for flow optimization and related policies. The recent advancement in the Internet of Things has created a keen interest in the research community as well as industry to integrate SDN in IoT networks. Similarly, virtualization in terms of networks, functions, and devices has also seen significant contributions in the recent past. In this article, we have reviewed both SDN and virtualization techniques for IoT and classified them into different types of solutions. SDN is limited to virtualizing the network layer of the stack where the IoT network traffic flow is optimized. Mostly the solutions aim at providing SDN services to resources constrained devices, enabling configuration services, or address security threats. Some works have involved function virtualization to implement common network functions in the logical domain. An important factor to note is that the future of IoT will not only be limited to SDN and isolated virtual functions. The later part of the article emphasizes software-defined IoT, which is a comprehensive solution, by incorporating controllers for different purposes in the control layer. This also integrates orchestration of virtual functions, as part of the vertical control layer. Additionally, we have presented several future research directions in this regard.

REFERENCES

[1] Muhammad Tahir Abbas, Afaq Muhammad, and Wang-Cheol Song. 2019. SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach. *J. Amb. Intell. Hum. Comput.* 11 (May 2019). DOI: https://doi.org/10.1007/s12652-019-01319-w

[2] Godfrey Anuga Akpakwu, Bruno J. Silva, Gerhard P. Hancke, and Adnan M. Abu-Mahfouz. 2018. A survey on 5G networks for the Internet of Things: Communication technologies and challenges. *IEEE Access* 6 (2018), 3619–3647. DOI: https://doi.org/10.1109/access.2017.2779844

[3] Mohammed Al-Shaboti, Ian Welch, Aaron Chen, and Muhammed Adeel Mahmood. 2018. Towards secure smart home IoT: Manufacturer and user network access control framework. In *Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA'18)*. IEEE, 892–899. DOI: https://doi.org/10.1109/aina.2018.00131

[4] Mamdouh Alenezi, Khaled Almustafa, and Khalim Amjad Meerja. 2019. Cloud based SDN and NFV architectures for IoT infrastructure. *Egypt. Inf. J.* 20, 1 (2019), 1–10. DOI: https://doi.org/10.1016/j.eij.2018.03.004

[5] Angelos-Christos Anadiotis, Laura Galluccio, Sebastiano Milardo, Giacomo Morabito, and Sergio Palazzo. 2019. SD-WISE: A software-defined WIreless SEnsor network. *Comput. Netw.* 159 (Aug. 2019), 84–95. DOI: https://doi.org/10.1016/j.comnet.2019.04.029

[6] Saleh Asadollahi, Bhargavi Goswami, and Mohammed Sameer. 2018. Ryu controller's scalability experiment on software defined networks. In *Proceedings of the 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC'18)*. IEEE, 1–5. DOI: https://doi.org/10.1109/icctac.2018.8370397

[7] L. Atzori, J. L. Bellido, R. Bolla, G. Genovese, A. Iera, A. Jara, C. Lombardo, and G. Morabito. 2019. SDN&NFV contribution to IoT objects virtualization. *Comput. Netw.* 149 (Feb. 2019), 200–212. DOI: https://doi.org/10.1016/j.comnet.2018.11.030

[8] Marc Balon and Bernard Liau. 2012. Mobile virtual network operator. In *Proceedings of the 2012 15th International Telecommunications Network Strategy and Planning Symposium (NETWORKS'12)*. IEEE, 1–6. DOI: https://doi.org/10.1109/netwks.2012.6381694

[9] Josep Batalle, Jordi Ferrer Riera, Eduard Escalona, and Joan A. Garcia-Espin. 2013. On the implementation of NFV over an OpenFlow infrastructure: Routing function virtualization. In *Proceedings of the 2013 IEEE SDN for Future Networks and Services (SDN4FNS'13)*. IEEE, 1–6. DOI: https://doi.org/10.1109/sdn4fns.2013.6702546

[10] Narmeen Zakaria Bawany, Jawwad A. Shamsi, and Khaled Salah. 2017. DDoS attack detection and mitigation using SDN: Methods, practices, and solutions. *Arab. J. Sci. Eng.* 42, 2 (1 Feb. 2017), 425–441. DOI: https://doi.org/10.1007/s13369-017-2414-5

[11] Z. Benomar, D. Bruneo, S. Distefano, K. Elbaamrani, et al. 2018. Extending openstack for cloud-based networking at the edge. In *Proceedings of the IEEE International Conference on Internet of Things (iThings'18) and IEEE Green Computing and Communications (GreenCom'18) and IEEE Cyber, Physical and Social Computing (CPSCom'18) and IEEE Smart Data (SmartData'18)*. 162–169. DOI: https://doi.org/10.1109/Cybermatics_2018.2018.00058

[12] Samaresh Bera, Sudip Misra, Sanku Kumar Roy, and Mohammad S. Obaidat. 2018. Soft-WSN: Software-defined WSN management system for IoT applications. *IEEE Syst. J.* 12, 3 (Sep. 2018), 2074–2081. DOI: https://doi.org/10.1109/jsyst.2016.2615761

[13] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang. 2019. PoBT: A light weight consensus algorithm for scalable IoT business blockchain. *IEEE IoT J.* (2019), 1–13. https://ieeexplore.ieee.org/document/8926457.

[14] S. Biswas, K. Sharif, F. Li, B. Nour, and Y. Wang. 2019. A scalable blockchain framework for secure transactions in IoT. *IEEE IoT J.* 6, 3 (Jun. 2019), 4650–4659. DOI: https://doi.org/10.1109/JIOT.2018.2874095

[15] Nikos Bizanis and Fernando A. Kuipers. 2016. SDN and virtualization solutions for the Internet of Things: A survey. *IEEE Access* 4 (2016), 5591–5606. DOI: https://doi.org/10.1109/access.2016.2607786

[16] Michel S. Bonfim, Kelvin L. Dias, and Stenio F. L. Fernandes. 2019. Integrated NFV/SDN architectures. *Comput. Surv.* 51, 6 (2019), 1–39. DOI: https://doi.org/10.1145/3172866

[17] A. Boudi, I. Farris, M. Bagaa, and T. Taleb. 2019. Assessing lightweight virtualization for security-as-a-service at the network edge. *IEICE Transactions on Communications* E102.B, 5 (2019), 970–977. DOI: https://doi.org/10.1587/transcom.2018EUI0001

[18] Joseph M. Bradley. 2013. The Internet of Everything: Creating Better Experiences in Unimaginable Ways. Retrieved March 12, 2019 from https://blogs.cisco.com/digital/the-internet-of-everything-creating-better-experiences-in-unimaginable-ways.

[19] R. Bruschi, P. Lago, G. Lamanna, C. Lombardo, and S. Mangialardi. 2016. OpenVolcano: An open-source software platform for fog computing. In *Proceedings of the 28th International Teletraffic Congress*, Vol. 2. 22–27. DOI: https://doi.org/10.1109/ITC-28.2016.212

[20] Peter Bull, Ron Austin, Evgenii Popov, Mak Sharma, and Richard Watson. 2016. Flow based security for IoT devices using an SDN gateway. In *Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud'16)*. IEEE, 157–163. DOI: https://doi.org/10.1109/ficloud.2016.30

[21] Farah Chahlaoui, Mohammed Raiss El-Fenni, and Hamza Dahmouni. 2019. Performance analysis of load balancing mechanisms in SDN networks. In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security (NISS'19)*. ACM, 36:1–36:8 pages. DOI: https://doi.org/10.1145/3320326.3320368

[22] Shaibal Chakrabarty, Daniel W. Engels, and Selina Thathapudi. 2015. Black SDN for the Internet of Things. In *Proceedings of the 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 190–198. DOI: https://doi.org/10.1109/mass.2015.100

[23] Hung-Li Chen and Fuchun Joseph Lin. 2019. Scalable IoT/M2M platforms based on kubernetes-enabled NFV MANO architecture. In *Proceedings of the 2019 International Conference on Internet of Things (iThings'19) and IEEE Green Computing and Communications (GreenCom'19) and IEEE Cyber, Physical and Social Computing (CPSCom'19), and IEEE Smart Data (SmartData'19)*. IEEE, 1106–1111. DOI: https://doi.org/10.1109/ithings/greencom/cpscom/smartdata.2019.00188

[24] N. M. M. K. Chowdhury and R. Boutaba. 2009. Network virtualization: State of the art and research challenges. *IEEE Commun. Mag.* 47, 7 (Jul. 2009), 20–26. DOI: https://doi.org/10.1109/mcom.2009.5183468

[25] Jacob H. Cox, Joaquin Chung, Sean Donovan, Jared Ivey, Russell J. Clark, George Riley, and Henry L. Owen. 2017. Advancing software-defined networks: A survey. *IEEE Access* 5 (2017), 25487–25526. DOI: https://doi.org/10.1109/access.2017.2762291

[26] Ala' Darabseh, Mahmoud Al-Ayyoub, Yaser Jararweh, Elhadj Benkhelifa, Mladen Vouk, and Andy Rindos. 2015. SDSecurity: A software defined security experimental framework. In *Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW'15)*. IEEE, 1871–1876. DOI: https://doi.org/10.1109/iccw.2015.7247453

[27] Stephen Dawson-Haggerty, Jorge Ortiz, Jason Trager, David Culler, and Randy H. Katz. 2012. Energy savings and the "Software-Defined" building. *IEEE Des. Test Comput.* 29, 4 (Aug. 2012), 56–57. DOI: https://doi.org/10.1109/mdt.2012.2202566

[28] Yasemin Demiral and Mehmet Demirci. 2018. An investigation of hypervisor effect on virtual networks performance. In *Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU'18)*. IEEE, 1–4. DOI: https://doi.org/10.1109/siu.2018.8404837

[29] Abhijeet Desai, K. S. Nagegowda, and T. Ninikrishna. 2016. A framework for integrating IoT and SDN using proposed OF-enabled management device. In *Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT'16)*. IEEE, 1–4. DOI: https://doi.org/10.1109/iccpct.2016.7530127

[30] Sadia Din, M. Mazhar Rathore, Awais Ahmad, Anand Paul, and Murad Khan. 2017. SDIoT: Software defined Internet of Thing to analyze big data in smart cities. In *Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops'17)*. IEEE, 175–182. DOI: https://doi.org/10.1109/lcn.workshops.2017.84

[31] Abebe Abeshu Diro, Haftu Tasew Reda, and Naveen Chilamkurti. 2018. Differential flow space allocation scheme in SDN based fog computing for IoT applications. *J. Amb. Intell. Hum. Comput.* (Jan. 2018). DOI: https://doi.org/10.1007/s12652-017-0677-z

[32] Ping Du, Pratama Putra, Shu Yamamoto, and Akihiro Nakao. 2016. A context-aware IoT architecture through software-defined data plane. In *Proceedings of the 2016 IEEE Region 10 Symposium (TENSYMP'16)*. IEEE, 315–320. DOI: https://doi.org/10.1109/tenconspring.2016.7519425

[33] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman. 2011. Network Configuration Protocol (NETCONF). Retrieved from https://tools.ietf.org/html/rfc6241.

[34] Pal Evensen and Hein Meling. 2009. SenseWrap: A service oriented middleware with sensor virtualization and self-configuration. In *Proceedings of the 2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'09)*. IEEE, 261–266. DOI: https://doi.org/10.1109/issnip.2009.5416827

[35] I. Farris, J. B. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin. 2017. Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems. In *Proceedings of the 2017 IEEE Conference on Standards for Communications and Networking (CSCN'17)*. 169–174.

[36] Ivan Farris, Tarik Taleb, Yacine Khettab, and Jaeseung Song. 2019. A survey on emerging SDN and NFV security mechanisms for IoT systems. *IEEE Commun. Surv. Tutor.* 21, 1 (2019), 812–837. DOI: https://doi.org/10.1109/comst.2018.2862350

[37] Nick Feamster, Jennifer Rexford, and Ellen Zegura. 2014. The road to SDN. *ACM SIGCOMM Comput. Commun. Rev.* 44, 2 (Apr. 2014), 87–98. DOI: https://doi.org/10.1145/2602204.2602219

[38] S. Fichera, M. Gharbaoui, P. Castoldi, B. Martini, and A. Manzalini. 2017. On experimenting 5G: Testbed set-up for SDN orchestration across network cloud and IoT domains. In *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft'17)*. IEEE, 1–6. DOI: https://doi.org/10.1109/netsoft.2017.8004245

[39] FLARE 2019. FLARE Networks. Retrieved March 12, 2019 from http://flare-networks.com/.

[40] Olivier Flauzac, Carlos Gonzalez, Abdelhak Hachani, and Florent Nolot. 2015. SDN based architecture for IoT and improvement of the security. In *Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 688–693. DOI: https://doi.org/10.1109/waina.2015.110

[41] Gartner 2017. IoT Statistics. Retrieved March 12, 2019 from https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016l.

[42] Carlos Gonzalez, Salim Mahamat Charfadine, Olivier Flauzac, and Florent Nolot. 2016. SDN-based security framework for the IoT in distributed grid. In *Proceedings of the 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech'16)*. IEEE, 1–5. DOI: https://doi.org/10.1109/splitech.2016.7555946

[43] Sghaier Guizani. 2017. Internet-of-things (IoT) feasibility applications in information centric networking system. In *Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC'17)*. IEEE, 2192–2197. DOI: https://doi.org/10.1109/iwcmc.2017.7986623

[44] Akram Hakiri, Pascal Berthou, Aniruddha Gokhale, and Slim Abdellatif. 2015. Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications. *IEEE Commun. Mag.* 53, 9 (Sep. 2015), 48–54. DOI: https://doi.org/10.1109/mcom.2015.7263372

[45] Evangelos Haleplidis, Jamal Hadi Salim, Joel M. Halpern, Susan Hares, Kostas Pentikousis, Kentaro Ogawa, Weiming Wang, Spyros Denazis, and Odysseas Koufopavlou. 2015. Network programmability with ForCES. *IEEE Commun. Surv. Tutor.* 17, 3 (2015), 1423–1440. DOI: https://doi.org/10.1109/COMST.2015.2439033

[46] Ana Belen Garcia Hernando, Antonio Da Silva Farina, Luis Bellido Triana, Francisco Javier Ruiz Pinar, and David Fernandez Cambronero. 2017. Virtualization of residential IoT functionality by using NFV and SDN. In *Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE'17)*. IEEE, 90–91. DOI: https://doi.org/10.1109/icce.2017.7889240

[47] Almulla Hesham, Fragkiskos Sardis, Stan Wong, Toktam Mahmoodi, and Mallikarjun Tatipamula. 2017. A simplified network access control design and implementation for M2M communication using SDN. In *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference Workshops*. IEEE, 1–5. DOI: https://doi.org/10.1109/wcncw.2017.7919082

[48] Peng Hu. 2015. A system architecture for software-defined industrial Internet of Things. In *Proceedings of the 2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB'15)*. IEEE, 1–5. DOI: https://doi.org/10.1109/icuwb.2015.7324414

[49] HUAWEI 2015. Huawei EC-IoT Solution. Retrieved March 12, 2019 from https://e.huawei.com/us/solutions/technical/sdn/agile-iot.

[50] Nachikethas A. Jagadeesan and Bhaskar Krishnamachari. 2014. Software-defined networking paradigms in wireless networks: A survey. *Comput. Surv.* 47, 2 (Nov. 2014). DOI: https://doi.org/10.1145/2655690

[51] Yaser Jararweh, Mahmoud Al-Ayyoub, Ala' Darabseh, Elhadj Benkhelifa, Mladen Vouk, and Andy Rindos. 2015. SDIoT: A software defined based internet of things framework. *J. Amb. Intell. Hum. Comput.* 6, 4 (01 Jun. 2015), 453–461. DOI: https://doi.org/10.1007/s12652-015-0290-y

[52] Anish Jindal, Gagangeet Singh Aujla, and Neeraj Kumar. 2019. SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment. *Comput. Netw.* 153 (Apr. 2019), 36–48. DOI: https://doi.org/10.1016/j.comnet.2019.02.002

[53] Anwar Kalghoum, Sonia Mettali Gammar, and Leila Azouz Saidane. 2017. Towards a novel forwarding strategy for named data networking based on SDN and bloom filter. In *Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA'17)*. IEEE, 1198–1204. DOI: https://doi.org/10.1109/aiccsa.2017.38

[54] Kubra Kalkan and Sherali Zeadally. 2018. Securing Internet of Things with software defined networking. *IEEE Commun. Mag.* 56, 9 (Sep. 2018), 186–192. DOI: https://doi.org/10.1109/mcom.2017.1700714

[55] Kotaro Kataoka, Saurabh Gangwar, and Prashanth Podili. 2018. Trust list: Internet-wide and distributed IoT traffic management using blockchain and SDN. In *Proceedings of the 2018 IEEE 4th World Forum on Internet of Things (WF-IoT'18)*. IEEE, 296–301. DOI: https://doi.org/10.1109/wf-iot.2018.8355139

[56] Pradeeban Kathiravelu, Leila Sharifi, and Luís Veiga. 2015. Cassowary. In *Proceedings of the 2nd Workshop on Middleware for Context-Aware Applications in the IoT (M4IoT'15)*. ACM, 1–6. DOI: https://doi.org/10.1145/2836127.2836132

[57] Imran Khan, Fatna Belqasmi, Roch Glitho, Noel Crespi, Monique Morrow, and Paul Polakos. 2016. Wireless sensor network virtualization: A survey. *IEEE Commun. Surv. Tutor.* 18, 1 (2016), 553–576. DOI: https://doi.org/10.1109/comst.2015.2412971

[58] JeongGil Ko, Byung-Bog Lee, Kyesun Lee, Sang Gi Hong, Naesoo Kim, and Jeongyeup Paek. 2015. Sensor virtualization module: Virtualizing IoT devices on mobile smartphones for effective sensor data management. *Int. J. Distrib. Sens. Netw.* 11, 10 (2015), 1–10. DOI: https://doi.org/10.1155/2015/730762

[59] Hadi Razzaghi Kouchaksaraei, Tobias Dierich, and Holger Karl. 2018. Pishahang: Joint orchestration of network function chains and distributed cloud applications. In *Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft'18)*. IEEE, 344–346. DOI: https://doi.org/10.1109/netsoft.2018.8460134

[60] Hadi Razzaghi Kouchaksaraei and Holger Karl. 2019. Service function chaining across OpenStack and kubernetes domains. In *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems (DEBS'19*). ACM Press, 240–243. DOI: https://doi.org/10.1145/3328905.3332505

[61] Kubernetes 2019. What is Kubernetes. Retrieved September 28, 2019 from https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/.

[62] Cheng Li, Zhengrui Qin, Ed Novak, and Qun Li. 2017. Securing SDN infrastructure of IoT–Fog networks from MitM attacks. *IEEE IoT J.* 4, 5 (Oct. 2017), 1156–1164. DOI: https://doi.org/10.1109/jiot.2017.2685596

[63] Hongda Li, Feng Wei, and Hongxin Hu. 2019. Enabling dynamic network access control with anomaly-based IDS and SDN. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (SDN-NFVSec'19*). ACM, 13–16. DOI: https://doi.org/10.1145/3309194.3309199

[64] Jie Li, Eitan Altman, and Corinne Touati. 2015. A general SDN-based IoT framework with NVF implementation. *ZTE Commun.* 13, 3 (2015), 42–45.

[65] Yong Li and Min Chen. 2015. Software-defined network function virtualization: A survey. *IEEE Access* 3 (2015), 2542–2553. DOI: https://doi.org/10.1109/access.2015.2499271

[66] Yuhong Li, Xiang Su, Jukka Riekki, Theo Kanter, and Rahim Rahmani. 2016. A SDN-based architecture for horizontal Internet of Things services. In *Proceedings of the 2016 IEEE International Conference on Communications*. IEEE, 1–7. DOI: https://doi.org/10.1109/icc.2016.7511053

[67] Jiaqiang Liu, Yong Li, Min Chen, Wenxia Dong, and Depeng Jin. 2015. Software-defined internet of things for smart urban sensing. *IEEE Commun. Mag.* 53, 9 (Sep. 2015), 55–63. DOI: https://doi.org/10.1109/mcom.2015.7263373

[68] Tie Luo, Hwee-Pink Tan, and Tony Q. S. Quek. 2012. Sensor OpenFlow: Enabling software-defined wireless sensor networks. *IEEE Comm. Lett.* 16, 11 (Nov. 2012), 1896–1899. DOI: https://doi.org/10.1109/lcomm.2012.092812.121712

[69] Taras Maksymyuk, Stepan Dumych, Mykola Brych, Dimas Satria, and Minho Jo. 2017. An IoT based monitoring framework for software defined 5G mobile networks. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*. ACM, 105:1–105:4. DOI: https://doi.org/10.1145/3022227.3022331

[70] Pedro Martinez-Julia and Antonio F. Skarmeta. 2014. Empowering the Internet of Things with software defined networking. *FP7 European Research Project on the Future Internet of Things* (2014). Retrieved October 10, 2019 from https://iot6.eu/sites/default/files/imageblock/IoT6 - SDN - IoT.pdf.

[71] Philippe Massonet, Laurent Deru, Amel Achour, Sebastien Dupont, Louis-Marie Croisez, Anna Levin, and Massimo Villari. 2017. Security in lightweight network function virtualisation for federated cloud and IoT. In *Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud*. IEEE, 148–154. DOI: https://doi.org/10.1109/ficloud.2017.43

[72] Prodromos Mekikis, Kostas Ramantas, Luis Sanabria-Russo, Jordi Serra, Angelos Antonopoulos, David Pubill, Elli Kartsakli, and Christos Verikoukis. 2020. NFV-enabled experimental platform for 5G tactile Internet support in industrial environments. *IEEE Transactions on Industrial Informatics* 16, 3 (2020), 1895–1903. DOI: https://doi.org/10.1109/tii.2019.2917914

[73] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. 2016. Network function virtualization: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutor.* 18, 1 (2016), 236–262. DOI: https://doi.org/10.1109/comst.2015.2477041

[74] D. Montero, M. Yannuzzi, A. Shaw, L. Jacquin, A. Pastor, R. Serral-Gracia, A. Lioy, F. Risso, C. Basile, R. Sassu, M. Nemirovsky, F. Ciaccia, M. Georgiades, S. Charalambides, J. Kuusijarvi, and F. Bosco. 2015. Virtualized security at the network edge: A user-centric approach. *IEEE Commun. Mag.* 53, 4 (2015), 176–186.

[75] Aki Nakao. 2012. Flare: Open Deeply Programmable Network Node Architecture. *Lecture Notes*. Retrieved March 12, 2019 from http://netseminar.stanford.edu/seminars/10_18_12.pdf.

[76] Stefan Nastic, Sanjin Sehic, Duc-Hung Le, Hong-Linh Truong, and Schahram Dustdar. 2014. Provisioning software-defined IoT cloud systems. In *Proceedings of the 2014 International Conference on Future Internet of Things and Cloud*. IEEE, 288–295. DOI: https://doi.org/10.1109/ficloud.2014.52

[77] Anne H. H. Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, and Michael Z. Sheng. 2016. IoT middleware: A survey on issues and enabling technologies. *IEEE IoT J.* 4, 1 (2016), 1–20.

[78] Binh Nguyen, Nakjung Choi, Marina Thottan, and Jacobus Van der Merwe. 2017. SIMECA: SDN-based IoT mobile edge cloud architecture. In *Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management*. IEEE, 503–509.

[79] Kim-Khoa Nguyen and Mohamed Cheriet. 2016. Virtual edge-based smart community network management. *IEEE Internet Comput.* 20, 6 (Nov. 2016), 32–41. DOI: https://doi.org/10.1109/mic.2016.127

[80] Mehdi Nobakht, Vijay Sivaraman, and Roksana Boreli. 2016. A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In *Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES'16)*. IEEE, 147–156. DOI: https://doi.org/10.1109/ares.2016.64

[81] Lukasz Ogrodowczyk, Bartosz Belter, and Marc LeClerc. 2016. IoT ecosystem over programmable SDN infrastructure for smart city applications. In *Proceedings of the 2016 5th European Workshop on Software-Defined Networks (EWSDN'16)*. IEEE, 49–51.

[82] Mike Ojo, Davide Adami, and Stefano Giordano. 2016. A SDN-IoT architecture with NFV implementation. In *Proceedings of the 2016 IEEE Globecom Workshops (GC Wkshps'16)*. IEEE, 1–6. DOI: https://doi.org/10.1109/glocomw.2016.7848825

[83] ONF TS-025 2015. OpenFlow Switch Specifications v1.5.1. Retrieved March 12, 2019 from https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf.

[84] OpenStack 2019. Open Source Software for Creating Private and Public Clouds.Retrieved September 28, 2019 from https://www.openstack.org/.

[85] Jianli Pan and James McElhannon. 2018. Future edge cloud and edge computing for Internet of Things applications. *IEEE IoT J.* 5, 1 (Feb. 2018), 439–449. DOI: https://doi.org/10.1109/jiot.2017.2767608

[86] Michael Pearce, Sherali Zeadally, and Ray Hunt. 2013. Virtualization: Issues, security threats, and solutions. *Comput. Surveys* 45, 2 (Feb. 2013). DOI: https://doi.org/10.1145/2431211.2431216

[87] Mehran Pourvahab and Gholamhossein Ekbatanifard. 2019. An efficient forensics architecture in software-defined networking-IoT using blockchain technology. *IEEE Access* 7 (2019), 99573–99588. DOI: https://doi.org/10.1109/ACCESS.2019.2930345

[88] Zhijing Qin, Grit Denker, Carlo Giannelli, Paolo Bellavista, and Nalini Venkatasubramanian. 2014. A software defined networking architecture for the Internet-of-Things. In *Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS'14)*. IEEE, 1–9. DOI: https://doi.org/10.1109/noms.2014.6838365

[89] Zhijing Qin, Luca Iannario, Carlo Giannelli, Paolo Bellavista, Grit Denker, and Nalini Venkatasubramanian. 2014. MINA: A reflective middleware for managing dynamic multinetwork environments. In *Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS'14)*. IEEE, 1–4. DOI: https://doi.org/10.1109/noms.2014.6838332

[90] Chao Qiu, F. Richard Yu, Fangmin Xu, Haipeng Yao, and Chenglin Zhao. 2018. Permissioned blockchain-based distributed software-defined industrial Internet of Things. In *Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps'18)*. IEEE, 1–7. DOI: https://doi.org/10.1109/glocomw.2018.8644520

[91] Gabriel Antonio F. Rebello, Igor D. Alvarenga, Igor J. Sanz, and Otto Carlos M. B. Duarte. 2019. BSec-NFVO: A blockchain-based security for network function virtualization orchestration. In *Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC'19)*. IEEE, 1–6. DOI: https://doi.org/10.1109/icc.2019.8761651

[92] Elisa Rojas, Roberto Doriguzzi-Corin, Sergio Tamurejo, Andres Beato, Arne Schwabe, Kevin Phemius, and Carmen Guerrero. 2018. Are we ready to drive software-defined networks? A comprehensive survey on management tools and techniques. *Comput. Surv.* 51, 2 (Feb. 2018). DOI: https://doi.org/10.1145/3165290

[93] Rishi Sairam, Suman Sankar Bhunia, Vijayanand Thangavelu, and Mohan Gurusamy. 2019. NETRA: Enhancing IoT security using NFV-based edge traffic analysis. *IEEE Sens. J.* 19, 12 (Jun. 2019), 4660–4671.

[94] Ola Salman, Imad Elhajj, Ali Chehab, and Ayman Kayssi. 2017. Software defined IoT security framework. In *Proceedings of the 2017 4th International Conference on Software Defined Systems (SDS'17)*. IEEE, 75–80. DOI: https://doi.org/10.1109/sds.2017.7939144

[95] Ola Salman, Imad Elhajj, Ayman Kayssi, and Ali Chehab. 2015. An architecture for the Internet of Things with decentralized data and centralized control. In *Proceedings of the 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA'15)*. IEEE, 1–8. DOI: https://doi.org/10.1109/aiccsa.2015.7507265

[96] Ola Salman, Imad Elhajj, Ayman Kayssi, and Ali Chehab. 2015. Edge computing enabling the Internet of Things. In *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT'15)*. IEEE, 603–608. DOI: https://doi.org/10.1109/wf-iot.2015.7389122

[97] Hunor Sandor, Bela Genge, and Gheorghe Sebestyen-Pal. 2015. Resilience in the Internet of Things: The software defined networking approach. In *Proceedings of the 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP'15)*. IEEE, 545–552. DOI: https://doi.org/10.1109/iccp.2015.7312717

[98] Pradip Kumar Sharma and Jong Hyuk Park. 2018. Blockchain based hybrid network architecture for the smart city. *Fut. Gener. Comput. Syst.* 86 (Sep. 2018), 650–655. DOI: https://doi.org/10.1016/j.future.2018.04.060

[99] Rob Sherwood, Glen Gibb, Kok-Kiong Yap, Guido Appenzeller, Martin Casado, Nick McKeown, and Guru Parulkar. 2009. Flowvisor: A network virtualization layer. *OpenFlow Switch Consortium, Technical Report* 1 (2009), 132.

[100] Seungwon Shin, Phillip A. Porras, Vinod Yegneswaran, Martin W. Fong, Guofei Gu, and Mabry Tyson. 2013. FRESCO: Modular composable security services for software-defined networks. In *Proceedings of 20th Annual Network & Distributed System Security Symposium (NDSS'13)*.

[101] Fatma Al Shuhaimi, Manju Jose, and Ajay Vikram Singh. 2016. Software defined network as solution to overcome security challenges in IoT. In *Proceedings of the 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO'16)*. IEEE, 491–496. DOI: https://doi.org/10.1109/icrito.2016.7785005

[102] Antonio Silva, Ana Garcia Hernando, and Mary Luz Mouronte. 2017. Residential wireless interfaces virtualization: A feasibility study. In *Proceedings of the 13th International Conference on Wireless and Mobile Communications (ICWMC'17)*. 67–72.

[103] Eugene Siow, Thanassis Tiropanis, and Wendy Hall. 2018. Analytics for the Internet of Things. *Comput. Surv.* 51, 4, Article 74 (Jul. 2018), 36 pages. DOI: https://doi.org/10.1145/3204947

[104] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Vijay Sivaraman, and Arun Vishwanath. 2016. Low-cost flow-based security solutions for smart-home IoT devices. In *Proceedings of the 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS'16)*. IEEE, 1–6. DOI: https://doi.org/10.1109/ants.2016.7947781

[105] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. 2015. Network-level security and privacy control for smart-home IoT devices. In *Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 163–167. DOI: https://doi.org/10.1109/wimob.2015.7347956

[106] M. Smith, M. Dvorkin, V. Laribi, V. Pandey, P. Gerg, and N. Weidenbacher. 2016. OPFlex Control Protocol. Retrieved from https://tools.ietf.org/html/draft-smith-opflex-03.

[107] Jungmin Son and Rajkumar Buyya. 2018. A taxonomy of software-defined networking (SDN)-enabled cloud computing. *Comput. Surv.* 51, 3, Article 59 (May 2018), 36 pages. DOI: https://doi.org/10.1145/3190617

[108] Daniel Steinberg and Stuart Cheshire. 2010. *Zero Configuration Network—The Definitive Guide*. O'Reilly Media.

[109] Sahrish Khan Tayyaba, Munam Ali Shah, Omair Ahmad Khan, and Abdul Wahab Ahmed. 2017. Software defined network (SDN) based Internet of Things (IoT). In *Proceedings of the International Conference on Future Networks and Distributed Systems (ICFNDS'17*). ACM, Article 15, 8 pages. DOI: https://doi.org/10.1145/3102304.3102319

[110] Mauro Tortonesi, James Michaelis, Alessandro Morelli, Niranjan Suri, and Michael A. Baker. 2016. SPF: An SDN-based middleware solution to mitigate the IoT information explosion. In *Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC'16)*. IEEE, 435–442. DOI: https://doi.org/10.1109/iscc.2016.7543778

[111] Ricard Vilalta, Arturo Mayoral, David Pubill, Ramon Casellas, Ricardo Martínez, Jordi Serra, Christos Verikoukis, and Raul Muñoz. 2016. End-to-end SDN orchestration of IoT services using an SDN/NFV-enabled edge node. In *Proceedings of the Optical Fiber Communication Conference*. OSA, 1–3. DOI: https://doi.org/10.1364/ofc.2016.w2a.42

[112] Jiafu Wan, Baotong Chen, Muhammad Imran, Fei Tao, Di Li, Chengliang Liu, and Shafiq Ahmad. 2018. Toward dynamic resources management for IoT-based manufacturing. *IEEE Commun. Mag.* 56, 2 (2018).

[113] Jiafu Wan, Shenglong Tang, Zhaogang Shu, Di Li, Shiyong Wang, Muhammad Imran, and Athanasios Vasilakos. 2016. Software-defined industrial Internet of Things in the context of industry 4.0. *IEEE Sens. J.* 16, 20 (2016), 1–1. DOI: https://doi.org/10.1109/jsen.2016.2565621

[114] Di Wu, Dmitri I. Arkhipov, Eskindir Asmare, Zhijing Qin, and Julie A. McCann. 2015. UbiFlow: Mobility management in urban-scale software defined IoT. In *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM'15)*. IEEE, 208–216.

[115] Ke Xu, Xiaoliang Wang, Wei Wei, Houbing Song, and Bo Mao. 2016. Toward software defined smart home. *IEEE Commun. Mag.* 54, 5 (May 2016), 116–122. DOI: https://doi.org/10.1109/mcom.2016.7470945

[116] Tong Xu, Deyun Gao, Ping Dong, Hongke Zhang, Chuan Heng Foh, and Han-Chieh Chao. 2017. Defending against new-flow attack in SDN-based Internet of Things. *IEEE Access* 5 (2017), 3431–3443. DOI: https://doi.org/10.1109/access.2017.2666270

[117] Yiannis Yiakoumis, Kok-Kiong Yap, Sachin Katti, Guru Parulkar, and Nick McKeown. 2011. Slicing home networks. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Home Networks (HomeNets'11*). ACM, New York, NY, 1–6.

[118] H. Yin, H. Xie, T. Tsou, D. Lopez, P. Aranda, and R. Sidi. 2012. *SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains*. Technical Report. Retrieved from http://tools.ietf.org/id/draft-yin-sdn-sdni-00.txt.

[119] Changhoon Yoon, Taejune Park, Seungsoo Lee, Heedo Kang, Seungwon Shin, and Zonghua Zhang. 2015. Enabling security functions with SDN: A feasibility study. *Comput. Netw.* 85 (Jul. 2015), 19–35.

[120] Ma Yun. 2013. Huawei Agile Network: A Solution for the Three Major Problems Facing Traditional Networking. Retrieved March 12, 2019 from http://e.huawei.com/hk/publications/global/ict_insights/hw_314355/industry%20focus/HW_314358.

[121] Alejandro Molina Zarca, Jorge Bernal Bernabe, Ruben Trapero, Diego Rivera, Jesus Villalobos, Antonio Skarmeta, Stefano Bianchi, Anastasios Zafeiropoulos, and Panagiotis Gouvas. 2019. Security management architecture for NFV/SDN-aware IoT systems. *IEEE IoT J.* 6, 5 (Oct. 2019), 8005–8020.

[122] Wei Zhang, Guyue Liu, Wenhui Zhang, Neel Shah, Phillip Lopreiato, Gregoire Todeschi, K. K. Ramakrishnan, and Timothy Wood. 2016. OpenNetVM. In *Proceedings of the 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization (HotMIddlebox'16)*. ACM Press, 26–31. DOI: https://doi.org/10.1145/2940147.2940155

Received May 2019; revised January 2020; accepted January 2020