IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.DOI

Towards Software-Defined Networking-based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges and Prospects

SHAHBAZ SIDDIQUI[1], SUFIAN HAMEED[1], SYED ATTIQUE SHAH[2], (Member, IEEE), IJAZ AHMAD[3], (Member, IEEE), ADEL ANEIBA[2], DIRK DRAHEIM[4], (Member, IEEE), SCHAHRAM DUSTDAR[5], (Fellow, IEEE)

[1]Department of Computer Science, NUCES, Karachi, 75160, Pakistan.
[2]School of Computing and Digital Technology, Birmingham City University, Millennium Point, B4 7XG, Birmingham, United Kingdom
[3]VTT Technical Research Centre of Finland, 02044 Espoo, Finland
[4]Information Systems Group, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia
[5]Distributed Systems Group, Vienna University of Technology, Vienna, Austria

Corresponding author: Syed Attique Shah (e-mail: syed.shah2@bcu.ac.uk).

ABSTRACT Internet of Things (IoT) is characterized as one of the leading actors for the next evolutionary stage in the computing world. IoT-based applications have already produced a plethora of novel services and are improving the living standard by enabling innovative and smart solutions. However, along with its rapid adoption, IoT technology also creates complex challenges regarding the management of IoT networks due to its resource limitations (computational power, energy, and security). Hence, it is urgently needed to refine the IoT-based application's architectures to robustly manage the overall IoT infrastructure. Software-defined networking (SDN) has emerged as a paradigm that offers software-based controllers to manage hardware infrastructure and traffic flow on a network effectively. SDN architecture has the potential to provide efficient and reliable IoT network management. This research provides a comprehensive survey investigating the published studies on SDN-based frameworks to address IoT management issues in the dimensions of fault tolerance, energy management, scalability, load balancing, and security service provisioning within the IoT networks. We conducted a Systematic Literature Review (SLR) on the research studies (published from 2010 to 2022) focusing on SDN-based IoT management frameworks. We provide an extensive discussion on various aspects of SDN-based IoT solutions and architectures. We elaborate a taxonomy of the existing SDN-based IoT frameworks and solutions by classifying them into categories such as network function virtualization, middleware, OpenFlow adaptation, and blockchain-based management. We present the research gaps by identifying and analyzing the key architectural requirements and management issues in IoT infrastructures. Finally, we highlight various challenges and a range of promising opportunities for future research to provide a roadmap for addressing the weaknesses and identifying the benefits from the potentials offered by SDN-based IoT solutions.

INDEX TERMS Internet of Things (IoT), Software-Defined Networking (SDN), SDN-based IoT Management frameworks, Systematic Literature Review, Network Function Virtualization, OpenFlow, Middleware, Blockchain, Security management, Fault tolerance, Load balancing, Scalability, Energy management.

I. INTRODUCTION

THE Internet of Things (IoT) is one of the most popular innovations in the current paradigm of information and communication technology. The term IoT has emerged from connecting embedded objects/things to the Internet. IoT infrastructure consists of data, sensing objects, computing, and communications to form a global and dynamic network infrastructure [1]. A collection of smart devices such as Radio

Frequency Identification (RFID) tags, sensors, smartphones, wearable devices, etc., are interconnected and can be used as data collection and dissemination points. Researchers foresee a future where IoT devices in large numbers will be deployed around us and will generate enormous amounts of data without requiring the active involvement of users [2]. The generated data sets will be collected, analyzed, and reported in an understandable form for various applications [3]. Yet, the field of IoT is about to create more attraction to researchers in the coming years due to the emergence of new application areas that can further improve our living standards [4].

The application domains of IoT range from leisure and sports such as smart activity monitors, to critical infrastructure such as manufacturing, healthcare, smart grids, and smart cities. The driving forces behind these applications include the development in sensor technologies, mobile devices, cloud infrastructures, and access technology providers, to name a few. The result is that huge volumes of IoT generated data containing real-world sensor-based information has dramatically expanded the demand for computing and storage resources for the IoT ecosystems to provide useful information or services [5]. In the IoT ecosystems, real-time processing is the primary requirement. In groups of several hundred, thousands, or even millions, IoT systems can theoretically handle parallel requests, which is required by several types of applications that need quick responses [6].

Successful deployments of IoT require merging heterogeneous communication infrastructures, which involves integrating smart gateways to link IoT devices with the Internet. Lately, research efforts are leading towards interconnecting the IoT infrastructure with technologies such as cloud computing, edge/ fog computing, big data analytics, machine learning, etc., that complement the potential of IoT. Furthermore, the ever-evolving IoT technology requires ubiquitous connectivity to billions of heterogeneous devices such as sensors, cameras, RFID devices, etc. [7]. The result is that IoT networks are growing enormously in size, and highly complicated due to the heterogeneity of device, access networks and protocols. Therefore, the IoT network management has become an extremely difficult challenge [8], and the challenge will be further exacerbated in networks beyond 5G, i.e., 6G, due to the humongous growth of connected devices. These challenges have led researchers to propose novel IoT management solutions, for instance, for load balancing, energy management, security, scalability, and fault tolerance [9].

Software-defined networking (SDN), considered as a breakthrough in communication networks, offers solutions to the management challenges of IoT. SDN simplifies the network management by separating the network control from the data forwarding elements, and logically centralizing it to high-end servers. Thus, the SDN framework proposes a three tier approach having an application plane, a control plane, and a data forwarding plane. The control plane, also called the SDN controller, maintains a global visibility of the network state enabling it to monitor, prioritize and de-prioritize network traffic through programmable Application Programming Interfaces (APIs) from a central vintage point. Therefore, SDN has been adopted as one of the main network management framework for IoT networks [10], [11]. SDN aims to make the network architecture more agile, flexible, and smart that can dynamically adopt to run-time changes in the network environment [11], [12]. Since an IoT network is highly dynamic mainly due to its resource constraints such as battery and processing power, and storage capability, the network has to adopt to its unique requirements. Such agility can be achieved through programmable network APIs in SDN, which makes SDN the most favorable networking architecture [13]–[18]. Fig. 1 shows a typical SDN-based IoT architecture.

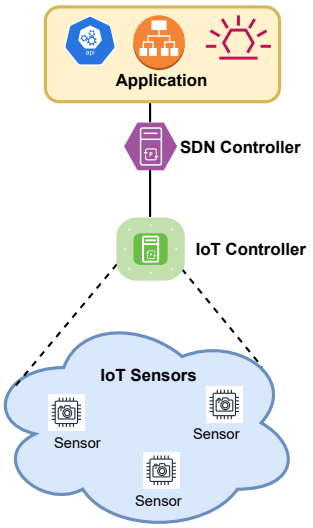

FIGURE 1. A general illustration of SDN-based IoT architecture.

Since the SDN framework greatly facilitates the management of IoT networks, substantial research efforts are dedicated in this direction. Several studies have been carried out to investigate different IoT reference architecture models based on SDN for current and potential IoT deployments. Therefore, in this article we survey the existing research efforts, fingerprint the research gaps, and shed light on how to overcome the existing challenges in this direction. We have systematically reviewed various SDN frameworks proposed for the IoT ecosystem. Moreover, we have included the published frameworks and have evaluated these frameworks to assess how they stack up in solving critical IoT management challenges in terms of provision of security services, fault tolerance, management of energy, load balancing, and scalability. In the following subsections, we present motivation behind this study, the related surveys published in the existing literature and the main contributions of this survey.

A. MOTIVATION

This survey is motivated by the realization that SDN tends to be a feasible alternative for IoT network architectures that

TABLE 1. A Comparison of Existing Surveys on IoT Frameworks Using SDN

Exiting Survey	Year	Fault tolerance	Load balancing	Security	Energy management	Scalability	SDIoT Framework Taxonomy
[19]	2013	✓		✓	✓		OpenFlow (OF), NFV, Middleware
[20]	2016			✓		✓	OF, NFV, Middleware
[21]	2016	✓	✓			✓	NFV
[22]	2016				✓		NFV
[23]	2017	✓	✓	✓			NFV, Middleware
[24]	2017	✓	✓	✓		✓	NFV, OF
[25]	2017				✓		NFV, OF
[26]	2018				✓		Middleware, OF
[27]	2018			✓	✓	✓	Middleware, NFV
[28]	2018	✓	✓			✓	Blockchain
[29]	2018		✓	✓		✓	Blockchain
[30]	2018	✓	✓			✓	Middleware
[31]	2018		✓			✓	Middleware
[32]	2018			✓			Middleware
[33]	2018			✓			Middleware, NFV
[34]	2018	✓		✓			Blockchain
[35]	2019			✓		✓	Blockchain
[36]	2019	✓	✓	✓		✓	Blockchain
[37]	2019		✓	✓		✓	Blockchain
[38]	2019		✓			✓	OF
[39]	2019				✓		Blockchain
[40]	2019			✓			Blockchain
[41]	2019				✓		Blockchain
[42]	2019			✓	✓		NFV
[43]	2019			✓			Middleware
[44]	2019			✓		✓	Blockchain
[45]	2020			✓			Blockchain
[46]	2020			✓		✓	Blockchain
[47]	2020		✓			✓	NFV, OF
[48]	2020				✓		NFV, OF
[49]	2020	✓	✓				Edge Computing, NFV
[50]	2020			✓			NFV
[51]	2020				✓		NFV
[52]	2020				✓		NFV, Blockchain
[53]	2020			✓	✓		Blockchain
[54]	2021			✓		✓	NFV
[55]	2021			✓		✓	Blockchain
[56]	2021	✓		✓		✓	NFV, Middleware
[57]	2021		✓	✓		✓	Middleware, Blockchain
[58]	2022	✓		✓		✓	Blockchain,NFV
[59]	2022	✓		✓		✓	NFV, Middleware, Blockchain
[60]	2022	✓	✓	✓		✓	NFV, Middleware
[61]	2022	✓	✓	✓	✓	✓	NFV, Middleware,Blockchain
Our Survey	2022	✓	✓	✓	✓	✓	NFV, Middleware, OF, Blockchain

enables optimization of the network and opens the possibility of developing new networks with more practical applications towards network management requirements. Although the notion of IoT-focused applications paints a beautiful picture of connected things with various applications, however, it does not come without a series of unique challenges. For IoT to become ubiquitous in industry and our everyday lives, these crucial challenges need to be tackled.

The combination of IoT and SDN (SDIoT) aims to connect objects over the internet by decoupling the control plane and the data plane. In the future, we envision that number of connected devices in IoT networks is in billions, and their management and control is a dynamic task that is a huge challenge for IoT networks. Without disturbing the basic architecture of existing implementations, SDN can render the IoT network scalable and programmable and provide potential solutions for the emphasized IoT management issues.

Recently, management for IoT networks has received attention as they are different from the traditional networks, which makes the conventional techniques and architecture inapplicable in the domain of IoT. The IoT network protocols and their legacy architecture have not been built to accommodate a large amount of data, mobility, and scalability. There are some drawbacks to the operation and management of these heterogeneous linked devices, which produce a massive amount of data. This rise in SDN adaptability has lead the initiative to use the same technique to manage IoT networks. Most recently, there are numerous efforts to utilize the potentials of the SDN paradigm to manage IoT networks. Several studies have been carried out to identify the IoT reference architecture models based on SDN for current and potential IoT deployments. The motivation behind our effort is to extensively review these existing SDN-based IoT management frameworks for exploring the unreaped opportunities and possible challenges. This survey aims to contribute to the knowledge of the design and implementation of SDN-

based IoT management frameworks and solution for various applications.

B. EXISTING SURVEYS

A number of surveys have been conducted during the last few years that broadly focus on various aspects of the IoT ecosystem using SDN. Table-1 shows a comparison between the existing research surveys on SDN based IoT management issues of IoT. Apart from these, a handful of research surveys have addressed the combined perspective of SDN-based IoT frameworks along with a few of their management issues [62]–[64], moreover, some surveys focus on only individual aspects of SDN-based IoT [65], [66]. Given that most of these existing surveys omit critical aspects and challenges of SDN-based IoT, hence, to the best of our knowledge, no survey has yet focused purely on SDN-based IoT frameworks keeping in view their management issues, i.e., fault tolerance, energy management, load balancing, security management, and scalability, provided that the integration of SDN and its evolving management challenges is a novel paradigm requiring high importance. In the following subsections, we illustrate the existing work in each of the identified SDN-based IoT management issues.

1) Fault Tolerance

In IoT networks, particularly in large-scale networks, it is theoretically impossible to operate when facing networking and other failures. Due to the SDN programmability, the network mechanism could be configured efficiently to attain fault tolerance and maintain the IoT networks on a large scale during failure [67]. In [68], Yu et al. present a detailed and systematic understanding and review of SDN reliability issues. It began with an introduction of SDN functionality, taking into account its current state of growth and offering an overview of SDN fault management solutions' two-dimensional taxonomy. In [63], Salman et al. in their survey critically analyze the solutions focused on SDN and fog computing to address IoT's key challenges in terms of fault-tolerant and scalability by highlighting the benefits and limitations of selected frameworks. In [69], Wang et al. discuss the techniques that accommodate benign faults and identify blockchain-based systems in which a fault-tolerant service replicates servers and coordinates client interactions with the aid of SDN flow tables.

2) Energy Management

SDN offers a better solution for green networking, which has become essential in network design and implementation for economic and environmental benefits [70]. It should be noted that, when introduced, security implementations in IoT increase energy consumption since security systems enact computations and communications that consume more power in the network [49], [71]. In [25] and [39], the authors address SDN/NFV-based security approaches. They also highlighted several advantages in scalability, on-demand network programmability, energy efficiency, and mobility. They also describes existing open SDN and NFV-related challenges for IoT security.

3) Load Balancing

In SDN, the controller views network resources globally combined with load optimization and applications' knowledge requirements. This approach makes SDN ideal to perform load balancing activities effectively and provides new possibilities in IoT networks for load balancing to boost the technology balance [72]. Also, to boost IoT network performance in multiple conscious routing approaches, load balancing technology is critical for the SDN networks. It is also used to systematically distribute the network's load to improve network capacity and quality of service (QoS). Therefore, with load balancing technology, the IoT network's overall efficiency can be significantly improved [71], [73].

4) Security Management

SDN was initially implemented to simplify the network configuration efforts in order to boost overall network performance, however, later SDN was found to be applicable to network security [74]–[76]. IoT networks are vulnerable to numerous security threats, some of which can not easily be identified. SDN is an evolving technology that can provide security protection solutions, because it is able to detect threats and respond faster than conventional networks, and all of this in an adaptive manner [74], [77].

5) Scalability

Due to the continuous changes in IoT networks, the focus needs to be renewed on security and privacy regarding data and users. Blockchain technology has emerged as a candidate for computerized transaction-based communications. The integration of IoT and blockchain technology offers various potential solutions in regards to scalability issues of IoT. Biwas et al. [78] highlighted various scalability issues and proposed the Lpeer network framework based on blockchain. The results obtained from their implementation prove that a scalable solution for IoT is applicable. In [37], the authors conducted a systematic review on blockchain's operations and classified their work into layers approach to highlight the blockchain-based solutions to the scalability issues in IoT.

C. SCOPE OF THIS SURVEY AND CONTRIBUTIONS

In this survey paper, we have systematically reviewed various SDN frameworks proposed for the IoT ecosystems with respect to various management issues. We have included published frameworks and have evaluated these frameworks to assess how they stack up in solving critical IoT management problems in terms of provision of security services, fault tolerance, management of energy, load balancing, and scalability. Our goal is to create a taxonomy and categorize existing SDN-based IoT frameworks. We have included frameworks that have been designed since 2010 and have evaluated these frameworks to assess how they stack up in solving critical IoT management problems in terms of

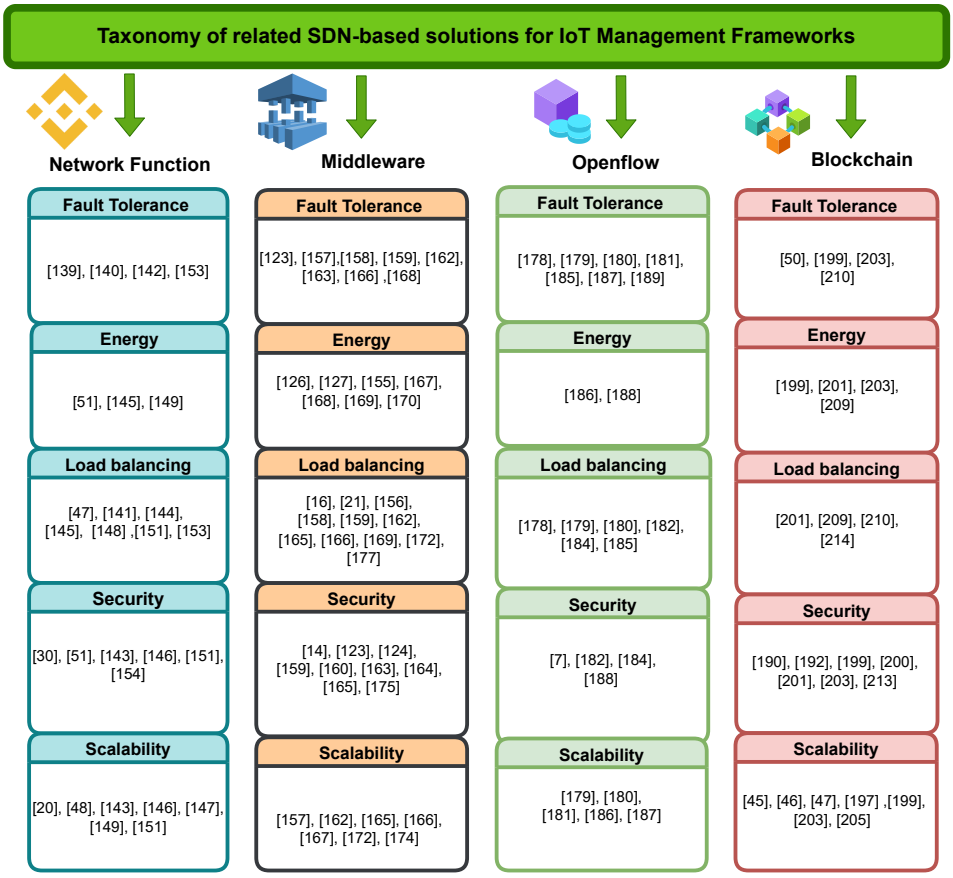

FIGURE 2. Taxonomy of related SDN-based solutions for IoT Management Frameworks.

provision of security services, fault tolerance, management of energy, load balancing, and scalability. We performed a Systematic Literature Review (SLR) based on Kitchenham's [79] well-known methodological framework to gather and analysis the existing research work. SLR is an evidence-based method to repetitively and impartially define, evaluate, and analyze all relevant evidence on a focused topic or research questions [80]. With the help of a predefined protocol, the SLR method selects and eliminates references and tests, and, ultimately, findings are synthesized by assessing specific studies and a clear proof of test questions. The main contributions of this survey are fourfold as following:

1) An SLR is conducted that provides an extensive review of existing SDN-based IoT (SDIoT) management frameworks published in reputable journals and conferences.
2) A tailored taxonomy is devised to categorize the related SDIoT solutions and a detailed discussion of each architecture is provided for better understanding of the current challenges.
3) The existing state-of-the-art SDN-based IoT management frameworks and solutions are classified and further investigated according to the following categories:
   i) Network Function Virtualization-based management
   ii) Middleware-based management
   iii) OpenFlow adaptation based management framework
   iv) Blockchain-based management
4) Every IoT management framework discussed in this paper has been analyzed with respect to its support of fault tolerance, security management, energy management, load balancing and scalabiltiy.
5) A set of critical research gaps that needs further investigation and research attention are identified to manage IoT networks more effectively.
6) Rising challenges and potential opportunities are highlighted to provide a road-map for future research directions to address the weaknesses of SDIoT solutions.

To the best of our knowledge, this is the first extensive survey of its kind to review all the current publications for SDN-based IoT solutions in terms of the full range of IoT implementation framework's management issues. Fig. 2 shows the derived taxonomy of existing studies categorized in accordance with various SDIoT management frameworks.

D. ORGANIZATION OF THE PAPER

The overall structure of this survey paper is shown in Fig. 3. Section II presents background knowledge of SDN and its working principles. Section III outlines the details for the

TABLE 2. List of Acronyms

Acronym	Description
6LPAN	Low-Power Wireless Personal Area Networks
AP	Application Plane
API	Application Programming Interface
BC	Blockchain
BRAS	Broadband remote access server
C1	Criteria-1
C2	Criteria-2
CLI	Command-Line interface
CoAP	Constrained Application Protocol
CP	Control Plane
DDS	Data Distribution Service
DISCO	Distributed OpenFlow controller
DOS	Denail of Service
DP	Data Plane
DPI	Deep Packet Inspection
ForCES	Forward and Control Element Separation protocol
HLR	Home Location Register
ILP	Integer Linear Programming
IoT	Internet of Things
IPSEC	Internet Protocol Security
LQE	Link Quality Estimation
M2M	Machine 2 Machine
MAC	Media Access Control
MANO	Management Orchestration
ML	Machine Learning
MME	Mobility Management Entity
MVNOS	Mobile Virtual Network Operators
NBI	Northbound-Interface
NE	Network Equipment
NFV	Network Function Virtualization
NMM	NFV Management Module
NOS	Network Operating System
NOX	OpenFlow controller
ODL	OpenFlow controller
OF	OpenFlow
ONIX	OpenDaylight
OPNFV	Open Network Function Virtualization
OS	Operating system
OVSDB	Open Virtual Switch Database Protocol
POX	Python based open source
QoS	Quality of Services
RFID	Radio Frequency Identification
RMM	Routing Management Module
SBI	Southbound-Interface
SDIoT	Software-Defined Internet of Things
SDN	Software-Defined Networking
SLR	Systematic Literature review
SSL	Secure socket layer
TCP	Transmission Control Protocol
UDP	User Data Protocol
VM	Virtual Machine
VNF	Virtual Network Function
WLM	White List Management
WSN	Wireless Sensor Network

SLR carried out for this study. Section IV covers a thorough discussion on the main IoT management challenges. In Section V, VI, VII, VIII, the NFV, Middleware, OpenFlow and Blockchain-based SDN management frameworks and their existing solutions are presented respectively along with their assessments regarding the defined research questions. In Sections IX, we summarized the outcomes of the survey with regards to the existing solutions and merger of different approaches that aid in addressing the IoT framework's management challenges. In Section X, we discuss the research challenges and future directions for the SDN-based IoT management frameworks in light of our survey. Finally, the conclusion of the paper is provided in Section XI. Table 2 describes the acronyms used in this research.

II. BACKGROUND

This section provides the required background knowledge of SDN and its architectural design by comparing it with traditional networking architecture.

A. SDN ARCHITECTURE

SDN is an evolving networking design architecture, construction architecture, and management architecture of the IoT ecosystem. SDN architecture consists of three layer of planes i.e., the DP, CP, and Application Plane (AP), as shown in Fig. 4. SDN architecture uses southbound and NBI API for communication with the DP and application plane with a protocol. OF is the most widely used protocol for this purpose [81].

1) Data Plane (DP)

The DP consists of network elements such as switches, routers, sensors nodes, etc. The DP is at the bottom of the SDN architecture and is responsible for managing data path and packets based on CP policies. According to the policies implemented by the CP, the DP forwards, drops and modifies packets [82]. Physical or virtual traffic routing and processing of network elements (NE)s such as switches, routers, and middleboxes are included in the DP [83]. Although data and CPs are implemented in the firmware of Network Equipment (NE) in traditional networking, the control functionalities are decoupled from the NE in SDN [84].

2) Control Plane (CP)

The software-based CP allows network resources and forwarding policies to be programmed and makes network management agile and versatile [85]. A logically centralized NOS or SDN controller is used to compose the CP [86]. Here, NOX, Python-based open source (POX), Floodlight, beacon controllers are the most commonly used controller [87]. The CP is responsible for configuring network elements with rules defined by the network applications designed on the top of controller [88]–[90]. Communication between applications (business logic and intelligence) and network devices is managed by the "brain" or the controller. The controller provides critical features such as storage of network topology,

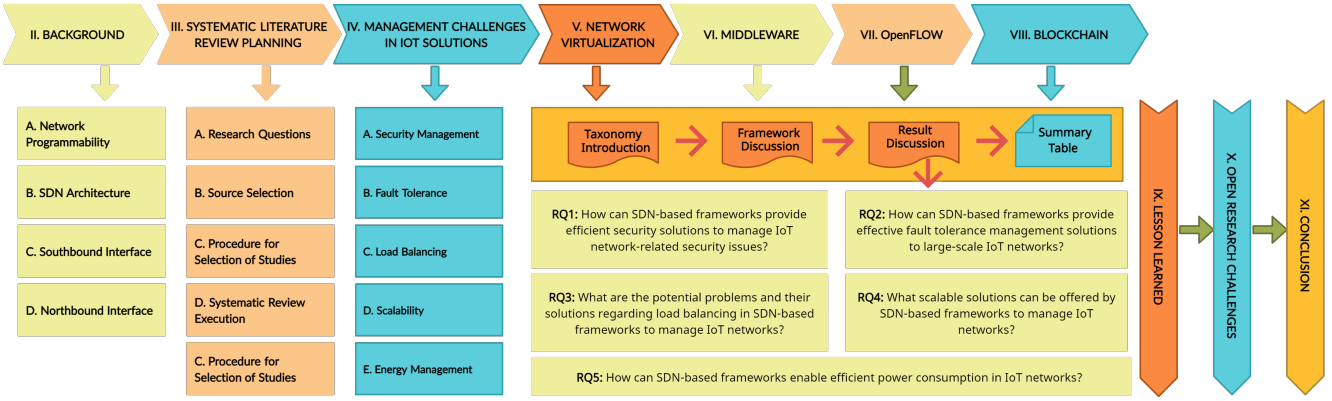

FIGURE 3. Overall Organization of the Survey Paper.

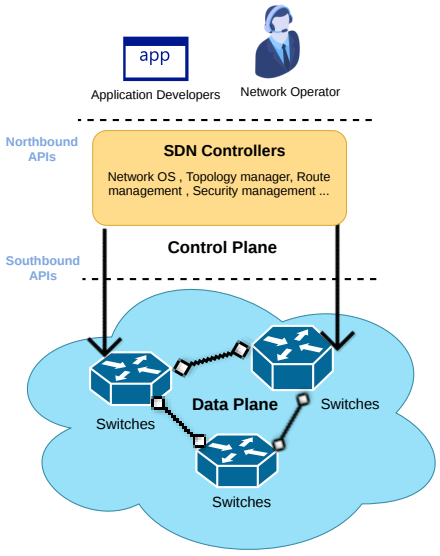

FIGURE 4. SDN-Based Network.

state data, alerts and system management, protection, and routing of the shortest paths [91]. These are the basic building blocks required by most network applications. The controller also abstracts the low-level specifics of the forwarding plane and offers the application plane an API called NBI [69].

3) Application Plane (AP)

The AP is the top layer, which contains numerous applications. It offers an end-to-end view of the entire network from a wide range of application domains such as military surveillance, health care or the smart transportation systems in which consumers or business applications live, to benefit from the resources available. It shares the control information with the SDN Controller via the Northbound interface (NBI) [69], [92], [93].

4) OpenFlow (OF)

OF is a programmable network interface protocol designed for controlling and monitoring all network devices. OF is considered to be one of the first SDN standards. Initially, it defined the communication protocol in SDN architectures that enabled the SDN controller to interact directly with the forwarding plane [94]. Using the OF protocol, a switch may be programmed to run identically to a legacy switch without re-configuring the switch manually if the network shifts [95]. A typical OF switch as shown in Fig. 5 contains a secure channel, flow table, and a group of tables. The group tables organize into multiple flow entries, which forward to a single identifier, identifying a node on the network. Such abstraction allows common output actions to be applied to flow entries, which can be changed efficiently. Incoming packets to the OF switch are compared with multiple flow table entries until a match is found, and a set of actions applicable for that particular flow entry is then performed [96].

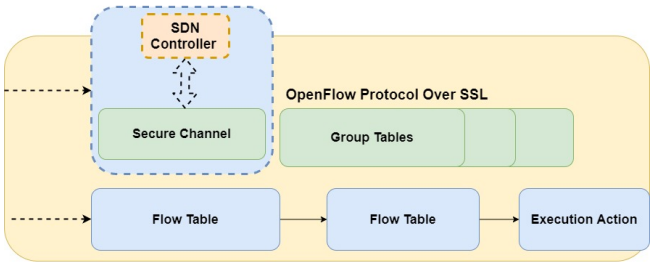

FIGURE 5. OpenFlow Switch Architecture.

5) Southbound Interface

The Southbound Interface (SBI) consists of the OF [97] and Forwarding and Control Element Separation (ForCES) [98] specifications that allows connectivity between controllers and switches and other network nodes with lower-level components or a DP layer. Southbound API enables the end-user to obtain better network control and encourage SDN controller performance levels to evolve based on real-time demands and needs. Moreover, the interface is an industry

norm that is justified by the perfect way the SDN controller can connect with the forwarding plane. To build a more flexible network layer for real-time traffic requirements, administrators may add or delete network switches and routers' internal flow tables.

6) Northbound Interface

The NBI's API provides communication between the SDN controller and the network applications with the help of automation stacks such as puppets, open packs, or open-source cloud pad [99]. SDN NBI's API integrates the SDN controller and the NBI API itself to incorporate more complicated frameworks such as firewalls, load balancers, and so on; and the controller will be responsible for ensuring that they communicate appropriately. NBI's API uses network computing paths, especially paths that comply with intended policies and computing paths that avoid loops, routing, and recovery from failures, and implementing protection policies. Table-3 shows the list of OF protocols for Southbound and NBI API for SDN controllers.

TABLE 3. OpenFLow APIs with SDN Controllers

Controllers	Architecture	SBIs	NBIs
DISCO [85]	Distributed	OF 1.0	REST
NOX [100]	Centralized	OF 1.0	ad-hoc
ONIX [100]	Centralized	OF 1.0, OvSDB	Onix-API
POX [101]	Centralized	OF 1.0	ad-hoc
FloodLight [102]	Centralized	OF 1.0 OvSDB, SNMP	REST, Java RPC
ONOS [103]	Centralized	OF 1.0,1.2	REST, Neutron

B. NETWORK PROGRAMMABILITY

Legacy network architectures rely on purpose-based and vendor-specific systems consisting of highly integrated and specialized forwarding chips [104], proprietary operating systems, and pre-defined features. An operator must configure each device using vendor-specific tools to enforce new network policies. Often, an operator needs to wait for a long time for including a new function before the device's manufacturer releases a software update that supports the intended component. Fig. 6 shows the main components of the legacy network architecture.

On the other hand, as a revolutionary paradigm, SDN allows network operators to be more flexible in managing and programming their network and in tackling their legacy network's shortcomings. SDN simplifies network management by separating the Control Plane (CP) from the Data Plane (DP) and making the network to be flexibly deployed and automatically configured by dynamically programming and reorganizing the network environment from the central SDN controller [105], [106].

SDN aims at making networking agile, flexible, and smart with the help of enhanced configuration, improved performance in network architecture and operations [25], [64], [107]. SDN provides network management orchestration as

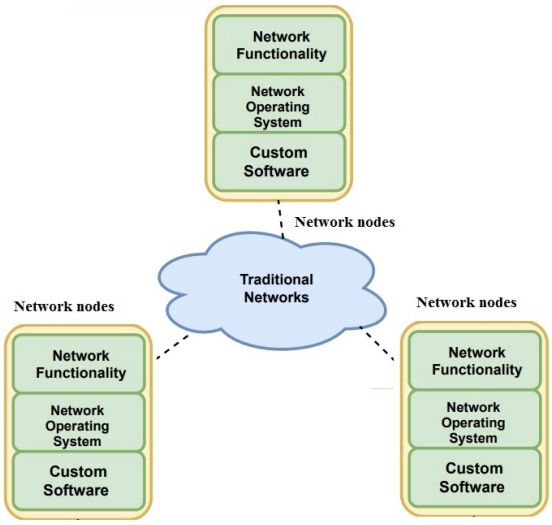

FIGURE 6. Legacy Network Architecture.

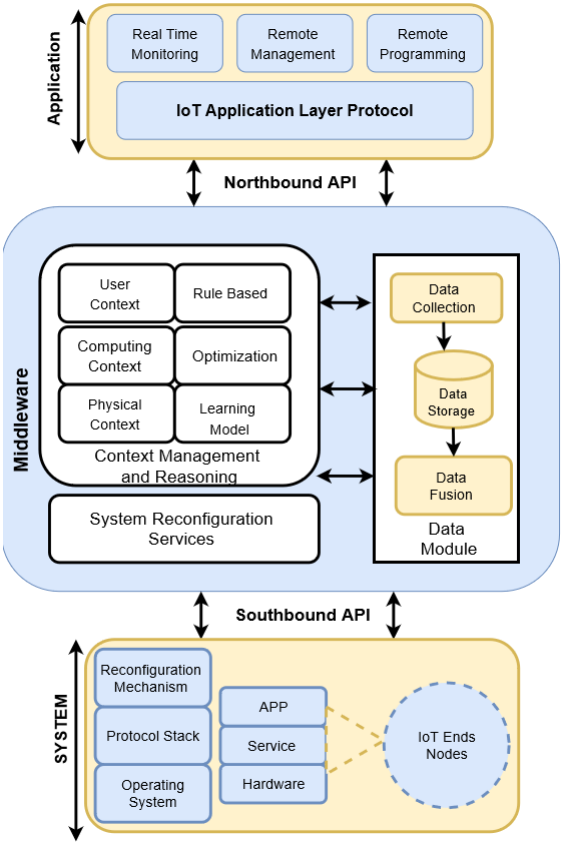

FIGURE 7. SDN Orchestration.

shown in Fig. 7. In [94], an OpenFlow (OF) switch concept was introduced even before the formal definition of SDN. To facilitate on-campus innovation networks, OF was developed by allowing researchers to test their ideas in an isolated 'slice' of the actual network [94]. By separating its CP and DP, this approach breaks the constraints of an "Ossified"

network structure. Gude et al. [108] proposed a network operating system named OpenFlow controller (NOX). NOX provides unified programming interfaces for the network (called NorthboundInterface (NBI)). The applications will take advantage of the network's logically centralized view using the NBIs provided by the Network Operating System (NOS). OF and NOX provide an effective solution for the SDN architecture principle (initially referred to as the NOX-based network).

III. SYSTEMATIC LITERATURE REVIEW PLANNING

This section outlines the overall plan for conducting the SLR for the study at hand. We will explain how the SLR was performed, including the research questions formalization, bibliographic source selection, and inclusion and exclusion criteria. The conducted SLR aims to provide the grounds for qualitative synthesis and information extraction leading towards finding the potential solutions to solving critical IoT management issues such as security service provisioning, fault tolerance, energy management, load balancing, and scalability through the available SDIoT frameworks. In this study, we primarily review the existing literature intending to systematically identify the current challenges and research opportunities for contributing to the knowledge-base of the SDN-based IoT framework.

A. RESEARCH QUESTIONS

This study aims to address the following primary research questions (RQ),

RQ1: How can SDN-based frameworks provide efficient security solutions to manage IoT network-related security issues?

IoT-based applications gather environment data and send it to central servers for review and processing. Maintaining privacy is essential in the application layer. Besides privacy, there are many other security issues, such as network routing attacks that can interrupt IoT services. Additionally, many IoT applications require trust management [109]. Therefore, IoT security monitoring is a crucial problem to be tackled. This question is about how SDN architecture provides IoT networks with protection efficiently.

RQ2: How can SDN-based frameworks provide effective fault tolerance management solutions to large-scale IoT networks?

Fault tolerance or reliability is the primary criteria for an IoT-based solution. SDN provides substantial reliability advantages. For example, due to global network visibility in SDN architecture, the CP can easily compose various network policies on the DP without conflicts. Several new features in SDN architecture still raise concerns about reliability. These features include the control DP separation architecture, which can increase network processing latency in the IoT network leading to network failures [68]. This question seeks to clarify the role of the SDIoT-based framework to provide efficient fault tolerance management in IoT networks and identify the challenges.

RQ3: What are the potential solutions regarding load balancing in SDN-based frameworks to manage IoT networks?

IoT network has limited network capacity to meet the quality of service requirements. One of the critical goals to maintain quality of service requirement is the load balancing problem, which helps spread data traffic among multiple resources to optimize network resources' efficiency and reliability [110]. This question seeks to clarify the role of the SDIoT-based framework to manage load balancing in IoT networks and identify the challenges and the techniques applied to guarantee the Quality of Service (QoS).

RQ4: What scalable solutions can be offered by SDN-based frameworks to manage IoT networks?

IoT infrastructure links together many sensors and devices for gathering information and sharing it with other applications through the Internet. It challenges the system's design and implementation to meet scalability and adaptability to the changing world and people's needs. Scalability means versatility that helps us to adequately address and satisfy the unique requirements when they arise. The main aim of making the system flexible is to meet evolving needs [111].

RQ5: How can SDN-based frameworks enable efficient power consumption in IoT networks?

IoT networks can achieve energy efficiency by increasing or decreasing data rates. Different sections of SDN-managed network dynamically configurable SDN framework to reduce power consumption. One way is to set the flow to the network traffic and bring unused devices into sleep mode. When traffic is poor, specific ports can be placed in sleep mode instead of the whole system. Another approach is to optimize or reduce the memory size used by forwarding switches as flow tables are stored in costly, power-hungry Ternary Content Addressable Memory (TCAM) [112].

B. SOURCE SELECTION

The selection of appropriate online bibliographic databases is essential to search primary studies and find proper evidence to address the research questions. In the following subsection, we will define the parameters used to select specific bibliographic sources and search strings. For bibliographic source selection criteria, we considered web articles' availability and the existence of advanced search mechanisms using keywords and content-based filtering (conference papers, journals, and magazines, etc.) and year of publication. We choose the following multidisciplinary electronic bibliographic databases: IEEE Xplore, Science Direct, Scopus, ACM Digital Library, and Springer Links.

Due to the integrative nature of the research questions, a variety of fitting search strings were required to be incorporated. To compose our search string, we considered keywords listed in Table-4, where each group is a keyword that either concatenates or not with another group string. We created search strings for two categories, as shown below in Equation (1) and (2), i.e., one for the survey findings and the other is to find the frameworks that are related to the research questions.

Here, $\wedge$ represents the logical AND, $\|$ represents the logical OR, and G represent the groups as shown in Table 4.

For finding related surveys that answers the research questions, we used the following equation for search string formation .

$$G1\{1 \parallel 2 \parallel 3 \parallel 4 \parallel 5\} \wedge G2 \wedge G3 \wedge G4\{i\} \wedge G5_{Survey} \quad (1)$$
$$i = \{NFV||Blockchain||Middleware||OpenFlow$$

For discovering the SDN-based IoT frameworks that answers the research questions, we have the following search string formation equation.

$$G1\{1 \parallel 2 \parallel 3 \parallel 4 \parallel 5\} \wedge G2 \wedge G3 \wedge G4\{i\} \wedge G5_{Framework} \quad (2)$$
$$i = \{NFV||Blockchain||Middleware||OpenFlow$$

TABLE 4. List of searching strings

G1	G2	G3	G4	G5
a. Load Balancing b. Fault Tolerance c. Energy Efficient d. Scalabilty e. Security	SDN	IoT	a. NFV b. Blockchain c. Middleware d. OpenFlow	a. Framework b. Survey

C. PROCEDURE FOR SELECTION OF STUDIES

The following inclusion and exclusion criteria were defined for the legitimacy of the primary gathered articles,

1. The primary study is an English-written article published in a scientific journal, conference proceeding, magazine, or book.
2. Publications in the shape of dissertations, in-progress research papers, guest editorials, posters, and blogs are excluded.
3. The primary study is published on or after the year 2010.
4. The primary study should clear the following three-phase selection and assessment process,

Phase i: An article will only be included in the following phase if it comes in the IoT and SDN domain and describing any of the following issues, i.e., energy management solution or design, fault tolerance, load balancing, and scalability and security. This stage focuses on the title, abstract, and the conclusion section.

Phase ii: An article will be included if it explains the proposed architecture design or evaluation of the proposed solution in detail. This stage evaluates full article content.

Phase iii: Selected paper screening is finalized and an article is removed unless it follows the following content requirements.

- C1: Does the selected paper fulfill any of the research questions or not?
- C2: Is the proposed architecture in the selected paper described in detail, and is it well-designed?

Each criterion (C1 and C2) has three possible responses, i.e., yes, partly, or no. "Yes" counts as 1 (one) point, "partly" counts as 0.5 points, and "no" counts as 0 (zero) point. An article must obtain a score equal to 2 (two) for selection, as defined in Equation (3):

$$C1 + C2 <= 2 \quad (3)$$

D. SYSTEMATIC REVIEW EXECUTION

The search for the required articles was carried out till the end of March 2022. Initially, we gathered a total of 668 research papers with the help of the defined search strings. We then began executing selection procedures, as defined in the primary study selection procedure, based on three stages of selection as defined in Section III. C. Having studied all abstracts and conclusions in phase 1 (screening phase), we select only those papers that provide SDN-based IoT solutions. We choose 328 research papers in this case and discarded 340. In phase 2, we selected articles explaining the SDN-based IoT management solution architecture based on inclusion and exclusion criteria and the relevance of the titles and keywords to the topic. After reviewing all the selected paper contents, we picked 224 studies and discarded 104 research papers. In phase 3, we discarded 68 more studies that did not meet the defined quality requirements based on the judgment criterion. In the last phase, the full-text screening of the selected papers was performed, and the papers were thoroughly analyzed by the authors. Moreover, with the help of forward and backward snowballing the number of inclusive studies increased from 156 to 188 resultant papers. Hence, after the final phase, the size of the selected paper database was 188 papers for the exploration of potential answers to the research questions. The detailed paper selection process during different phases are summarized in Table-5. The complete procedure from initial selection to full-text selection is summarized in Fig. 8.

To classify the selected articles' information, metadata forms were created to organize the details and considered annotations. The obtained metadata, containing information such as publication year, keywords, authors' names and affiliations, journal/conference name, research type, SDN and IoT architecture details, management issue details, etc., were coded for analysis to answer the research questions. The majority of the resulting papers were published between 2018 and 2021, indicating an increasing interest in how SDN can solve management problems in the IoT domain. In accordance with the research question, an initial classification was performed to show the number of survey and framework-based papers with respect to different challenges in various approaches, as shown in Fig. 9. Fig. 10 depicts the total number of survey and framework-based papers focusing on various IoT management challenges. With the available information on different IoT management challenges in various areas, in Fig. 11 we have also extracted the distribution of the identified papers in accordance with their research methods.

TABLE 5. Paper selection process during different phases

Digital Library	Initial Selection	Inclusion/ Exclusion Criteria	Title and Keywords	Abstract	Forward and Backward Snowballing	Full-Text Selection
IEEE	152	-43	-22	-16	+09	80
Science Direct	96	-31	-19	-14	+05	37
ACM	72	-24	-13	-09	+04	30
Springer	49	-19	-12	-08	+07	17
Scopus	299	-223	-38	-21	+07	24
Total	**668**	**-340**	**-104**	**-68**	**+32**	**=188**

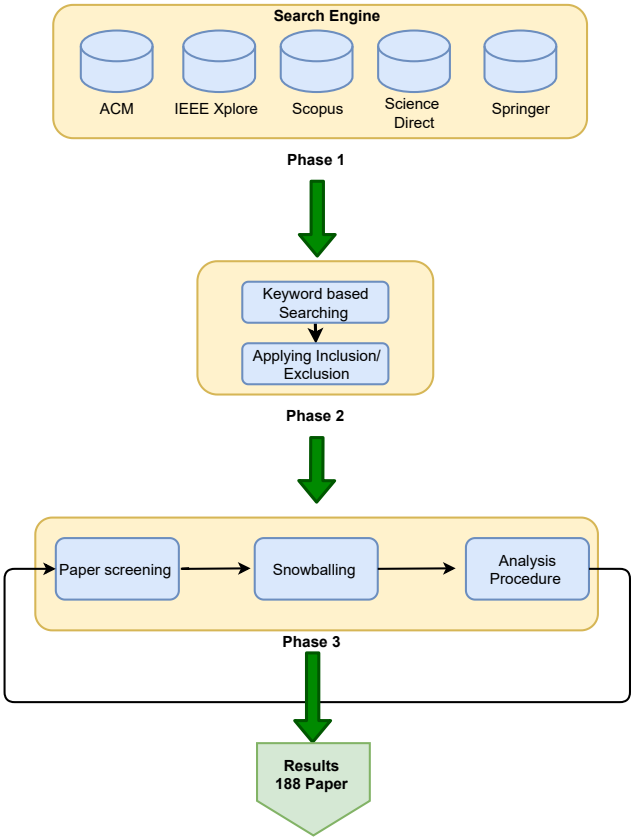

FIGURE 8. Search and selection process.

IV. MANAGEMENT CHALLENGES IN IOT SOLUTIONS

Conventionally managing a network, involves the use of a set of management protocols that facilitate the sharing of data between users and networks of all kinds [113]. Due to the wide range of networked systems found on the Internet today, controlled network modules can have diverse characteristics in terms of storage, processing capacities, and energy usage [114]. IoT network management should be able to provide functionalities, among other capabilities, such as to monitor network status, detect faults, configure operating parameters, collect network performance information, and manage its operation [115]. Moreover, due to the wide-spread Internet connectivity the management challenges faced by traditional Wireless Sensor Network (WSN) are now inherited to IoT domain as well [116]. These management challenges have been characterized by [117], [118], [119], [120] as,

1: Security management
2: Fault tolerance
3: Load balancing
4: Scalability
5: Energy management

The IoT network management solutions should be designed in a manner that provides a range of management functions that cater to the above-mentioned IoT management issues.

A. SECURITY MANAGEMENT

IoT network applications collect data from the sensors/devices and send it for analysis and processing to central servers. This data can vary from health specifics to purchasing habits and sales at a retailer. For companies, this data has monetary value. One critical issue during the whole process is maintaining privacy [121]. In addition to privacy, security concerns, such as network-based routing attacks and botnet attacks, can disrupt the IoT services [122] [123]. Furthermore, several IoT applications require trust management for reliable data fusion and enhanced information security [124]. Because of these mentioned reasons, IoT security management is critical to ensure the safety of networks and efficient data transmission. However, in IoT networks, the security functionality becomes even more difficult due to the heterogeneous nature of these networks equipped with resource constraints IoT devices [125]. Therefore, traditional IoT security systems are inefficient and require extensive adaptation, including overall IoT network framework redesigns. The new IoT network management frameworks require innovative mechanisms to deal with these unique challenges on security management. The need for more robust solutions is piling due to user unawareness, untimely device updates, lack of adequate security protocols for IoT authentication and IoT encryption.

B. FAULT TOLERANCE

Fault tolerance mechanisms in IoT networks address device failures and ensure that the network will continue to operate smoothly and reliably [126]. There are numerous reasons for failures to occur in the IoT networks. Device battery depletion is the most common reason for failures [127]. Also, inaccurate readings caused by various environmental and technical factors may propagate the devices. The multi-hop communication nature of IoT networks exacerbates a lot of

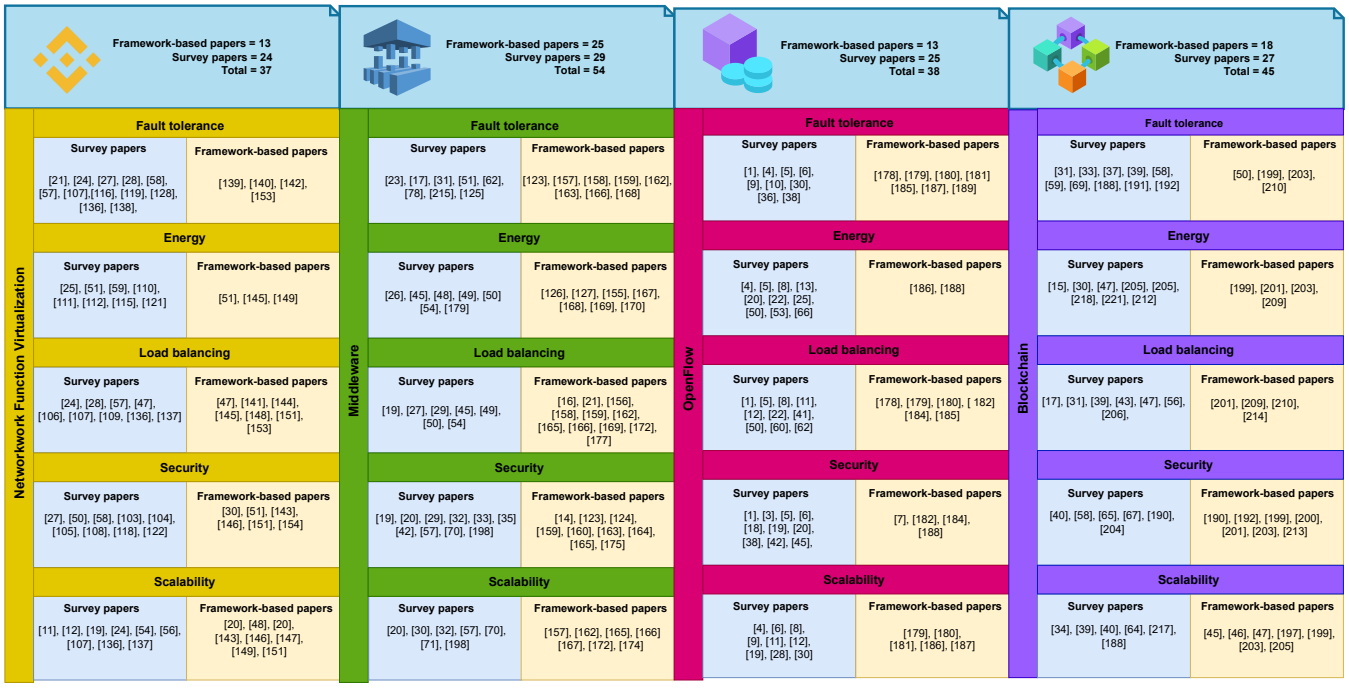

FIGURE 9. Number of Survey and Framework-based Papers in the considered areas.

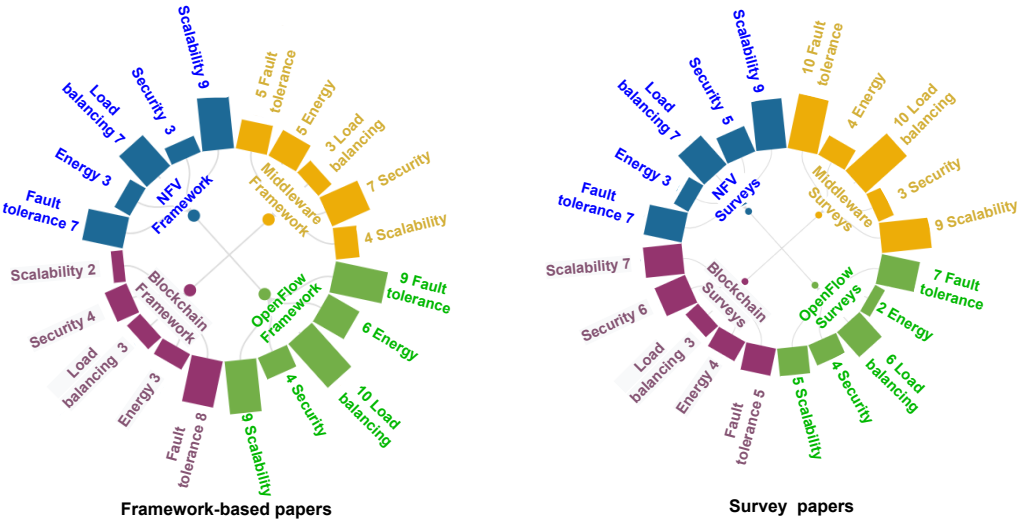

FIGURE 10. Number of papers identified in terms of IoT Management Challenges.

failures [128]. Moreover, the following failures can occur at all architectural levels of IoT applications,

- Sensor and actuator nodes may be absent.
- Network connections may be down.
- Processing and storage components may fail to operate correctly.

Therefore, IoT infrastructures must support state-of-the-art fault tolerance mechanisms to be able to recover from these malfunctions.

C. LOAD BALANCING

Load balancing is one of the essential strategies in IoT environments that aims to assign proper utilization of IoT infrastructure for optimizing the use of sensors or other connected devices. The role of load balancing in IoT networks is correlated with the number of connected objects employed for sharing data. The imbalance in the network traffic within the IoT network, which is hampered by resources, results in waste of resources [129]. As a result, load balancing within IoT networks leads to efficient use of resources within IoT networks. IoT networks can expand their life span through load balancing, which reduces the grid's energy consumption

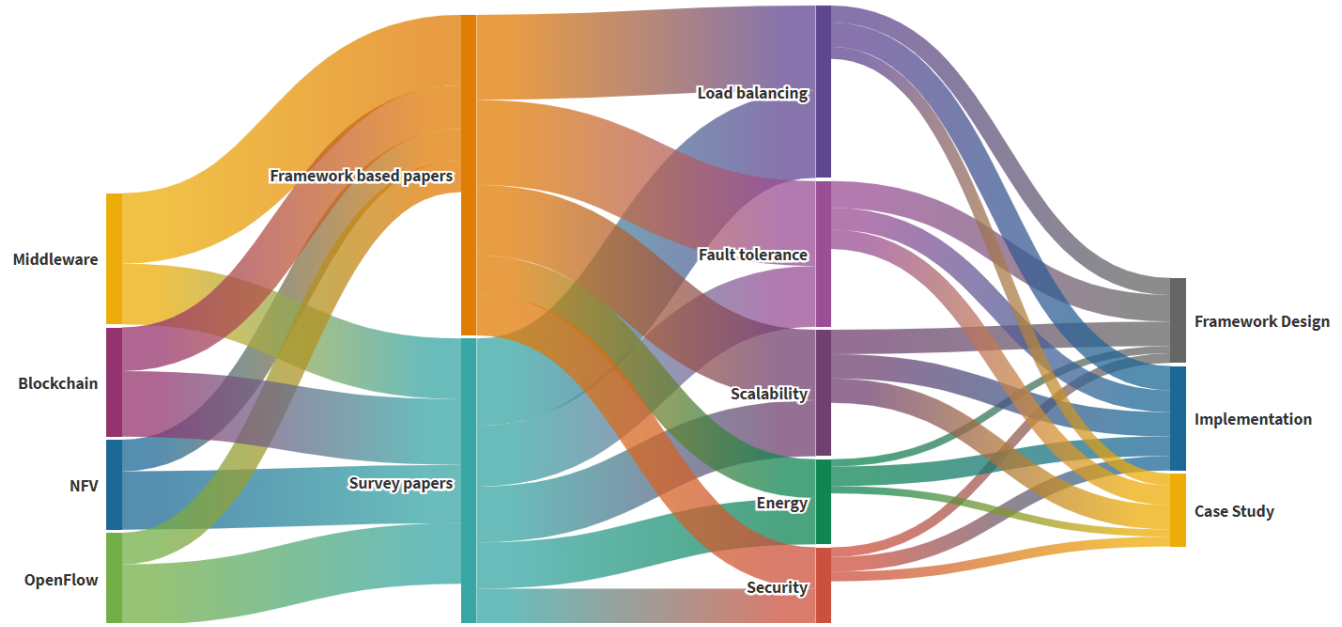

FIGURE 11. Distribution of papers according to their research methods.

[130]. The clustering in the network is one way to achieve load balancing in an IoT infrastructure. The IoT network is organized into clusters where the cluster's head coordinates and communicates within the nodes [131]. Network clustering reduces the routing table size, conserves network bandwidth, increases network lifetime, reduces redundant data packets, and decreases energy consumption [132] [133].

D. SCALABILITY

Scalability means versatility that allows one to adapt to the changes and grow with them and achieve specific needs when they arise. The main advantage of scalability is that it enables the system to operate gracefully without any undue delay and unproductive resources and makes fair use of the available resources. Any scheme that can manage the network with the rising amount of growth is a beneficial function. With the increasing definition of IoT in the future, scalability is a big challenge in IoT [134]. An IoT system connects several sensors, actuators, and other devices to enable information sharing and a large number of applications via the Internet. It challenges the design and the system's growth to meet scalability and adaptability to the people's evolving digital needs.

E. ENERGY MANAGEMENT

Inherently, IoT devices' energy is constrained because of the sensor nodes deployed in a remote area with no access to a permanent power source [135]. IoT network energy management is concerned with energy conservation within the network for the connected nodes. Over time the power of the existing battery shrinks, and the power depletion can not be readily replaced as the sensor nodes are remotely deployed. Duty cycling is one of the techniques used to preserve energy on IoT equipment. The devices will wake up during an intermittent time if necessary and sleep during this technique [136]. From this discussion, it is clear that a management solution for those networks should have an elaborate component of energy management in order to be able to work smoothly in an IoT network.

V. NETWORK FUNCTION VIRTUALIZATION BASED SDN MANAGEMENT FRAMEWORKS

Network Function Virtualization (NFV) offers an advantage for the ICT industry by separating the network hardware into a virtualized solution. The concept of switching functionality, routing assistance, and other components are now run in software applications such as virtual applications. These network functions are available in a group format from the remote location. Table-6 shows some renowned network functions of a network device, switching device, gateway device, and security devices. The key benefit of using NFV is that it enables eliminating middlelayers that are deployed in traditional networks for cost effectiveness and flexibility. Network and infrastructure features allow the use of a single physical platform by different providers, applications, and tenants [47]. On the other hand, NFV technology facilitates the coexistence of multi-tenancy as well.

TABLE 6. List of Virtual network functions of Network layer

Network devices	Virtual network function
Network security devices [30]	Firewall, DOS attack detection
Network switching devices [137]	NAT, BRAS, routers
Mobile Network devices [138]	HLR/HSS GPRS support
Tunning getways device [139]	IPSEC/SSL SLA

NFV infrastructure consists of two layers, i.e., the hard-

ware resources layer, and a virtualization layer, as shown in Fig. 12. The hardware resources layer is responsible for dealing with the storage and network services that include data centers, edge nodes for IoT domains, etc. The virtualization layer is accountable for providing virtual functions to the lower layer or hardware resource layer.

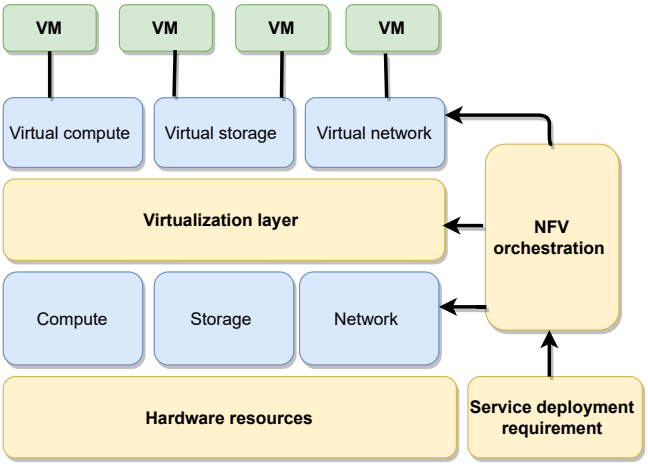

FIGURE 12. Virtual Network functions infrastructure.

The SDN NFV based architecture generally consists of three modules, i.e., control module, forwarding devices, and NFV platform, as shown in Fig. 13. In the control module, the SDN controller communicates with NFV orchestration with the help of the NBI-API interface to derive essential network functions from the NFV platform layer. Forwarding devices are responsible for forwarding the packets to the controller through an interface for decision-making. The NFV orchestration device is responsible for providing the virtualized network's functionality and is managed by standard interfaces by the SDN controller. It translates the requirements of the logic policy into optimized routing routes. The NFV orchestration system enforces task assignments [140].

Function Virtualization is implemented in a series of building blocks to define connectivity and to construct communication services between them through an NFV architecture, which uses various techniques to virtualize full network node functions [141]. The architecture of the NFV consists of three key (a) VNF: These are the software features responsible for carrying out basic network operations; (b) NFV Infrastructure (NFVI): This platform handles multiple VNFs, virtual storage, and processing; and (c) NFV Management and Orchestration (NFV-MANO): Offers an architectural framework for interfaces and referrals [142].

This section aims at answering the research questions based on NFV taxonomy with a combination of SDN frameworks to address the IoT management challenges. We will discuss the different SDN/NFV frameworks proposed in the existing literature to address the IoT management challenges and to identify future directions.

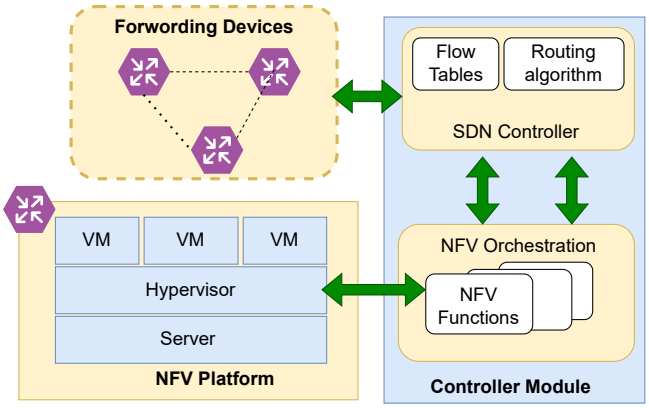

FIGURE 13. SDN-based NFV architecture providing the virtualized network's functionality.

A. INFRASTRUCTURE SERVICES NFV/SDN ARCHITECTURE

1) Motivation

An IoT network faces many challenges in cooperating with various network resources and providing services such as security, computing, power management, etc. IoT networks need to be tailored to the situation and provide the required services. SDN can provide network operations that provide control layer operations with the help of Network Functions Virtualization (NFV). SDN offers a resource management mechanism for IoT networks, thus helping infrastructure resources to be deployed effectively.

2) Proposed Framework

In a proposed architecture [143], authors have been influenced by SDN and virtualization network function capabilities for IoT infrastructure resources. The NFV and SDN make it easy to program network services. The NFV portion of Virtual Network Functions (VNF) shifts network functions from dedicated hardware to software. In NFV, SDN enables the complex establishment of relations between VNFs. The proposed architecture [143], as shown in Fig. 14, is composed of four different layers, i.e., (1) service layer, (2) global OS layer, (3) virtualization, and physical layer. The service layer incorporates all service-level functions. The global OS layer integrates cloud orchestration tools and SDN controllers. SDN controller layer is responsible for end-to-end network and IT resources management. It handles all network elements' dynamic configuration and re-configuration parameters. The virtualization layer organizes hardware resources on virtual machines made accessible to the layers above. Finally, the perception layer consists of IoT sensors responsible for extracting data and provided to the upper layer. The authors' aim in the proposed framework is to decouple hardware from network operations, minimizing resource management costs with the NFV and SDN's help. VNF services are transferred from dedicated applications through the use of SDN controllers.

3) Critical Analysis

The proposed architecture is very general and does not provide specific information regarding the various components' operations and relationships in different layers. Service level function and infrastructure resources definition is not presented. Furthermore, no specifics are given about how SDN and NFV collaborate to handle IoT.

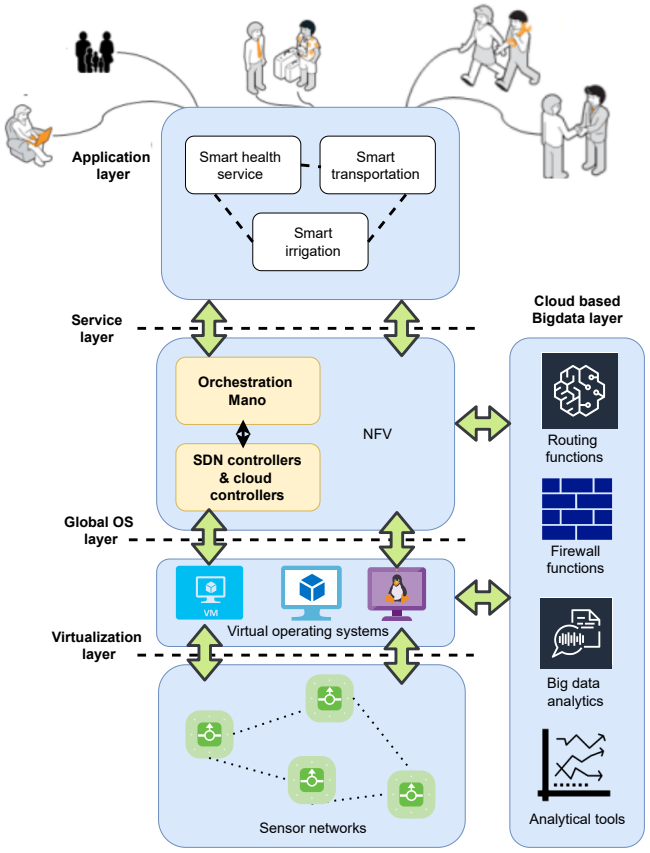

FIGURE 14. Infrastructure services NFV/SDN architecture.

B. SDN-BASED IOT FRAMEWORK USING NFV

1) Motivation

IoT nodes can have a high computing capacity with cloud computing support, but deploying cloud computing approaches to IoT poses challenges for the SDN research paradigm and the network virtualization integration feature. To create communication services, NFV virtualizes entire network functions that are then interlinked. Instead of making custom hardware equipment for network operations, network functions are virtualized by one or more virtual machines that execute heterogeneous processes. Li et al. [144] suggested that networking features such as routing, secure tunneling between IoT gateways, and prioritization of traffic for QoS in an IoT network can be implemented with OpenFlow-based SDN and NFV implementation.

2) Proposed Framework

As shown in Fig. 15, the authors in [144] proposed an IoT architecture based on SDN with NFV implementation. The proposed framework consists of the application, control, and infrastructure layer. The application layer includes IoT servers for various applications and services via API. The control layer comprises SDN controllers that are running on a distributed OS. The distributed OS provides logically centralized IoT control and viewing in a physically distributed network data forwarding environment. The infrastructure layer consists of IoT gateways and SDN switches for access to various IoT devices such as RFIDs and sensors via control Interface DP. Authors suggest that with OpenFlow-based SDN and NFV implementation, it will be possible to implement IoT networking functions such as routing, secure tunneling among IoT gateways, and prioritizing traffic for QoS in a centralized, programmable controller. Resourcefully distributed OS assists NFV-based SDN frameworks for IoT infrastructures. Distributed OS approach offers centralized control and view of heterogeneous IoT services.

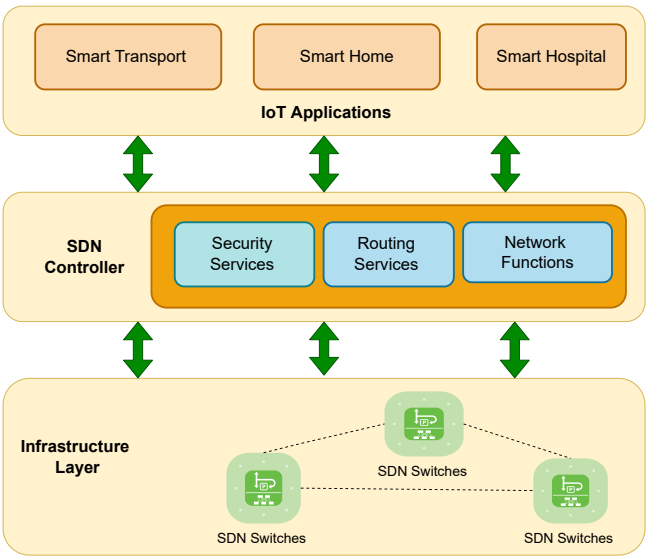

FIGURE 15. A typical SDN-based IoT framework with NFV.

3) Critical Analysis

As illustrated in Fig. 15, the proposed architecture is quite generic, and various layers are not detailed appropriately, as implementation and assessment details are lacking. Evaluations are necessary to understand the performance improvements made by the delivery of OS for IoT network management. Moreover, studies must be carried out to measure the overall cost resulting from the virtualization of the architecture network functions.

C. A DISTRIBUTED SECURE BLACK SDN IOT ARCHITECTURE

1) Motivation

The big concern in the IoT domain is confidentiality, safety, reliability, and network performance. With the support of centralized networks in collaboration with controllers, SDN can handle the IoT network assets with the help of integration with NFV.

2) Proposed Framework

Network Virtualization Feature incorporates the theme of using virtual machines that handle routing, switching, and other network operations instead of using specialized hardware. However, NFV needs to be monitored and coordinated. The SDN, therefore, comes with a solution to handle all virtual machines and networks by decoupling the CP and the DP. The IoT device is distributed in nature, and the sensor nodes keep sending data to the controller applications accompanied by environmental perception. This is why the SDNIoT environment's deployment has become more effective in low power consumption, efficiency enhancement, and security issues reduction. The authors in [145], presented Black SDN-IoT with NFV implementation for smart cities using NFV integration with the SDN controller, as shown in Fig. 16. The proposed architecture is based on the layered approach: the application layer, the CP, DP, and the perception layer. The application layer consists of multiple smart services of a smart city. In the CP, the virtualize function provide services to distributed SDN controller such as routing, security, resource management, etc. with the help of VNF. The DP is responsible for forwarding the data packet from the perception to the CP. SDN in the proposed architecture is distributed in nature according to its security roles. One of the distributed SDN controller's critical roles in preventing the dissipation of the data among the nodes by making them directed to themselves, thus saving energy by the process. The security controller controls the cluster domain and protects each cluster of SDN security controller against attacks produced within and outside the IoT network.

3) Critical Analysis

SDN-IoT has numerous unique challenges, and only a few researchers have tackled these challenges. The proposed centralized Black SDN-IoT architecture [145] with NFV is considered for smart cities for energy savings, load balancing, and network scalability purposes. The authors introduced several hierarchical SDN controllers to enhance availability, integrity, confidentiality, etc., in IoT network data. The authors do not present any detailed work on security modules in NFV or explore any algorithmic security approach. Implementation and analysis are also missing in the architecture.

D. ENHANCING IOT SECURITY USING NFV-BASED ANALYSIS

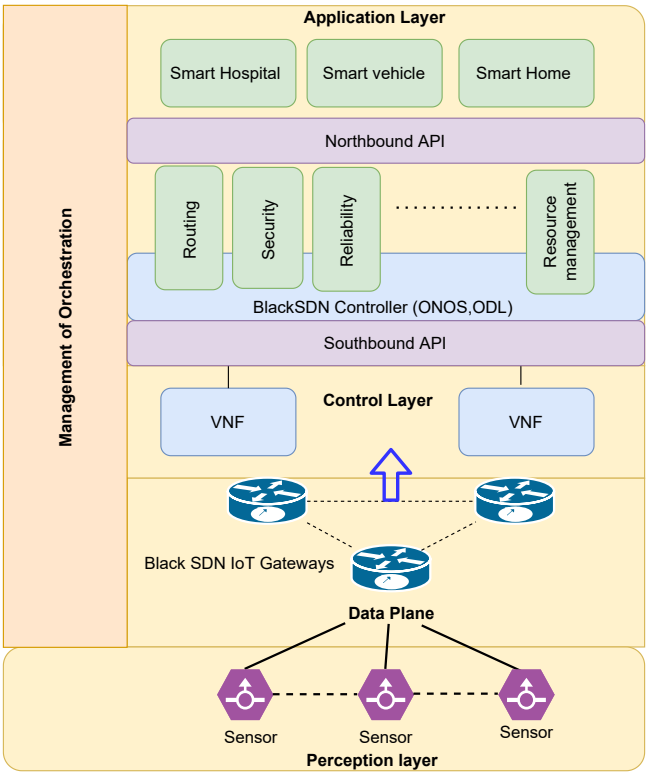

FIGURE 16. A distributed SDN-IoT architecture.

1) Motivation

With the evolution of IoT gadgets and their applications, we are moving toward the era of smart computing. The security of these smart gadgets is at high risk due to cyber-attacks. Conventional security mechanisms to manage IoT network security issues have limitations in terms of scalability and cost.

2) Proposed Framework

Authors in [146] proposed a Docker-based framework that deployed virtual network security functions at IoT gateway, as shown in Fig. 17. These virtual functions are stored in a cloud structure. IoT gateway is responsible for fetching these virtual network security functions from the cloud according to the requirements. These VNFs play an important role in improving the security of IoT environments. The proposed architecture based on docker technology consists of three layers, i.e., core network, IoT gateway, and IoT environment. The core network contains the Docker hub, which includes the repository for all docker images. The Docker images can be deployed from this layer with a docker pull command. IoT gateway layer represents an edge that hosts various dockers VNFs modules such as firewall, intrusion detection, SDN switches. The IoT environment contains the IoT nodes such as cameras, sensors, etc. The authors used an Open Platform as NFV, and OPNFV consists of different IoT nodes that utilize different network functions deployed from the upper layer of the OPNFV master.

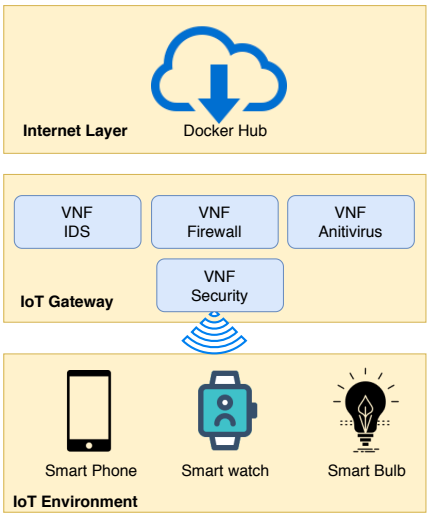

FIGURE 17. NETRA: Enhancing IoT Security using NFV.

3) Critical Analysis

The authors compare the two architectures VM-based (OP-NFV) without SDN and Docker-based (NETRA) based on SDN with performance indicators such as storage, memory, latency, network, and scalability. The result suggested that NFV as container-based virtualization with an SDN-based approach works better than the existing solution based on a VM-based framework. The proposed solution improved the security of the IoT environment containing nodes such as smart cameras, smart sockets using appropriate VNFS. The authors only focus on security features and have not discussed the workflow of these NFV based security functions.

E. ENERGY AWARE SDN/NFV ARCHITECTURE

1) Motivation

The IoT defines a new state of life where billions of IoT sensors link to colossal network traffic. A programmable network such as SDN can cope with such data explosion and resource constraints with the help of NFV, which also allows on-demand network deployment. SDN and NFV support each other for an IoT architecture where many network management challenges can be solved. The authors in [51], proposed an architecture that describes an Integer Linear Programming (ILP) problem to maximize IoT nodes' energy usage by enabling an appropriate number of NFV nodes and assigning optimal nodes to those starting NFV nodes.

2) Proposed Framework

As shown in Fig. 18, the authors' proposed SDN-based NFV solution for IoT network includes two modules: NFV Management Module (NMM) and RoutingManagement Module (RMM) [51]. NMM consists of a VNF container and VNF manager to preserve the available network function definitions and provide an API to an enabled NFV node. RMM node module maintains its neighbors' energy-state information and shares it with the controller. The controller uses energy-state information to enable an optimum number of NFV nodes and creates corresponding energy-aware routes maintained at RMM. The authors use (ILP) problem and map it into broad IoT networks to solve the energy consumption in IoT nodes. They proposed an algorithm in which each source node has two shortest routes to the accessible NFV nodes. Every route has a related energy cost (total contact energy). The algorithms assign the source node to one of the available NFV nodes based on energy and activation costs.

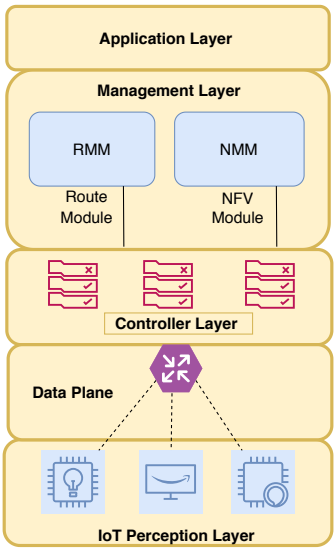

FIGURE 18. Energy-aware SDN/NFV architecture.

3) Critical Analysis

The authors implemented a proposed IoT network energy-sensitive SDN-based NFV architecture. They used a heuristic approach (EA-SDN/NFV) as the ILP problem is NP-complete to implement the proposed architecture. The results indicate that the proposed SDN-based NFV solution shows better results in terms of IoT node's energy consumption. However, it has some limitations, for example, the architecture is implemented with only 40 IoT nodes with grid topology in Cooja simulation. Similarly, battery existence is identical across all IoT nodes: a coin-type lithium-ion battery with 3V and 150 mA-h power rating. The suggested architecture focuses mainly on IoT nodes' energy usage, not on other features such as fault tolerance, security resource features.

F. NFV-BASED IOT SECURITY FOR HOME NETWORKS

1) Motivation

IoT networks are not powerful enough to detect malicious code and protect themselves against it. Billions of IoT devices (estimates vary from 10 to 50 billion by 2020) are fertile ground for various attacks such as DDoS, botnet attacks, etc., leading to terrorism, data theft, and other security concerns [147] [148]. The authors in [149] proposed a new method to defend multiple IoT devices through a single VNF

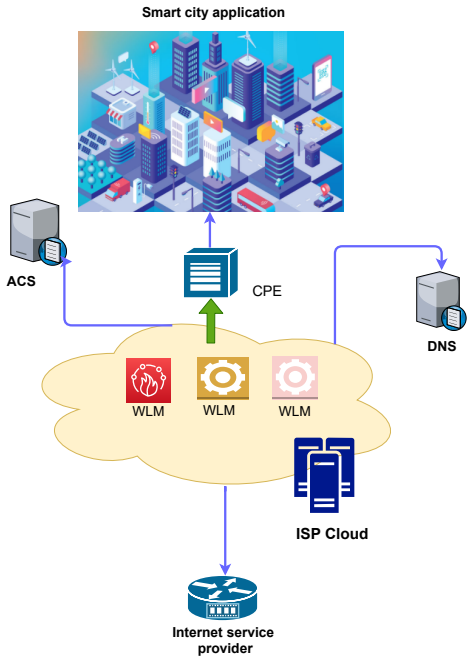

FIGURE 19. IoT security for home networks using NFV.

through the ISP network. The approach is based on the manufacturer's use definition (MUD), a Whitelist management (WLM) IoT protection scheme.

2) Proposed Framework

Afek et al. [149] proposed architecture as shown in Fig. 19. The author aims to ensure that all IoT application packets comply with the MUD file guidelines. That means that each packet passes to a MUD file for blocking or not blocking purposes. Thus, the MUD compliance present in the form of virtual network service is evaluated by WLM. WLM decides whether or not a packet passes a whitelist. The packet is either dropped or enabled in whitelist/ MUD compliance.

3) Critical Analysis

The proposed framework is implemented on an ISP network environment as a proof of concept. The data-plane is implemented using Open vSwitch (OVS) version 2.8.1 with OF 1.3. The CP runs as an application (in Python) over Ryu (Open-source OF controller) [149]. The implementation leverages OVS's caching capability, supporting the pipelined OVS and OF architectures, where packets cross several tables, each having numerous rules before being listed. The authors do not discuss the result of the proposed architecture in detail. Moreover, the architecture lacks modules for energy, security, and resource management.

G. CONTEXT-AWARE SDN-NFV-BASED IOT ARCHITECTURE

1) Motivation

In supporting data-oriented Internet-of-Things (IoT) applications, the current host-centered Internet infrastructure is inefficient, where contextual data packet information is desirable for in-network forwarding and processing. The authors in [150] proposed a context-aware IoT architecture that can forward and process IoT traffic in the DP to fill the gap between IoT and IP, based on contextual information

2) Proposed Framework

Du et al. [150] focuses on the prototyping of an IoT traffic management context-aware forwarding/processing mechanism. The contextual information is transmitted from both a sensor layer and an application layer to mitigate IoT network challenges related to scalability, discoverability, stability, reliability, computational, and battery limitations. The aim is to allow multiple Mobile Virtual Network Operators (MVNOS) over shared wireless infrastructures. Therefore, to enable SDN services for MVNOs, the architecture uses programmable switches. On FLARE platform, the IoT gateway program ensures trailer slicing. As shown in Fig. 20, the authors suggested a system designed with sliced MVNO networks. Data is collected by sensors (e.g., wearable devices) and then distributed via IoT gateways to MVNO networks. After managing the MVNO switches, the data collected will eventually be aggregated and analyzed by a central service controller. The authors to guarantee protection and privacy, data collection, and processing they segregated from the Internet they simultaneously run several different MVNO networks for various applications

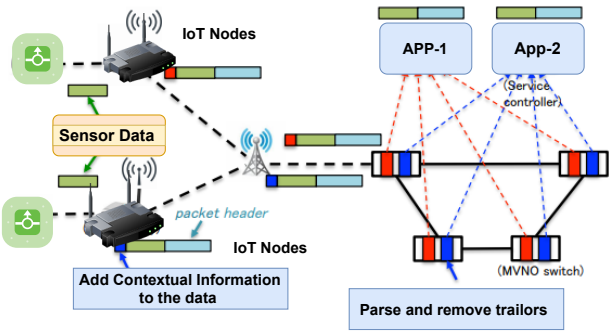

FIGURE 20. Context-aware SDN NFV based IoT architecture.

3) Critical Analysis

The proposed solution supports IoT heterogeneity through VNF, which is dynamically generated, modified, monitored, and removed according to the network situation's requirements. The focus is to provide functionalities such as discovery and connectivity of IoT devices, data collection and encapsulation, and forwarding/processing context-aware packets. The proposed framework runs on a very small testbed, and the viability of the proposed framework for handling large IoT networks is a question mark.

H. SECURITY IN LIGHTWEIGHT NETWORK FUNCTION

1) Motivation

Smart IoT applications enable many IoT devices and networks to be connected to various applications operating on fog and cloud computing platforms. Creating a federated virtual network is one solution to linking IoT devices with cloud and fog services. This strategy's primary advantage is that the IoT uses an application-specific federated network where no traffic from other applications passes, and devices may communicate with several remote services. Multiple cloud providers and IoT networks cover this federated network, but it can be operated as a single organization. Federated virtual networks can be managed centrally and protected from a security point of view, with a consistent global security strategy for IoT networks.

2) Proposed Framework

As shown in Fig. 21, the proposed architecture by [151] comprises 3 VNFs within the ETSI NFV architecture: a deep packet inspection engine (DPI), a firewall (FW), and an intrusion detection system (IDS). The VNF Manager is responsible for developing, upgrading the VNFs, and controlling the service feature chaining of the VNF Orchestrator, putting NFV within containers to reduce the hardware requirements on the edge router. To transport data from the IoT network controller to the cloud, the authors suggested in the proposed architecture that it is essential to translate the IoT data into a protocol that the cloud network can understand. First, the IoT gateway performs this translation, and then the information is sent by the IoT gateway into a cloud that gathers the data and performs higher-level processing and analysis by providing advanced network services such as FW and DPI.

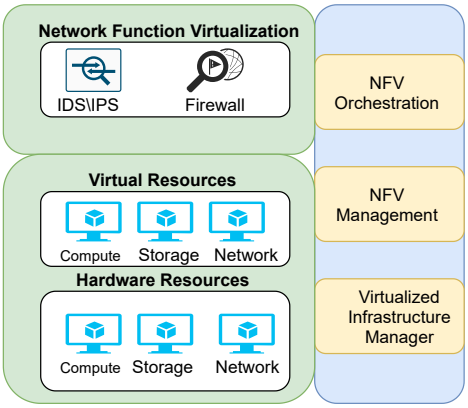

FIGURE 21. Security in Lightweight Network Function.

3) Critical Analysis

IoT-related security is the key objective of the proposed architecture. The concept is to use virtual networks to access cloud resources and federate various virtual networks to control and protect the federated network as a single, isolated entity. To implement the proposed security architecture, the NFV and SFC are focused on numerous IoT and cloud networks, a global network safety strategy. The authors presumed that each IoT and cloud platform has an NFV/SFC infrastructure used in each IoT and cloud platform in the federated network to deploy, configure and chain the protection VNF. The authors did not implement the proposed architecture for an application and mainly focus on security management rather than cover other management issues.

I. APPLICATION OF INTERNET OF THINGS SERVICE PLATFORM

1) Motivation

Fog computing and IoT technologies play a prominent role in smart city deployment, facilitating the sharing and management of urban knowledge. The authors in [152] suggest that fog computing-based SDIoT architecture can have the potential to effectively addresses big data processing and network scalability issues.

2) Proposed Framework

The authors suggested a fog-based computing and NFV platform for the IoT as shown in Fig. 22. Fog nodes are connected to base stations or routing devices by high-capacity fibers in this architecture, reducing end-to-end transmission delay. Fog nodes can also be installed on the edge of the cell network so that the same fog node can be used by various base stations or routing devices to process data. The fog nodes in the network can be linked to the cloud, allowing full use of the processing resources of the cloud to increase network deployment flexibility. The fog node will upload the data to the cloud for processing when there is a large amount of data to be processed on the network, and the fog node does not have sufficient computing power.

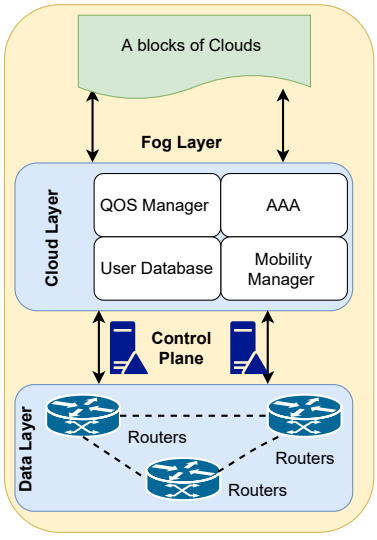

FIGURE 22. Application of IoT based on Fog Computing.

3) Critical Analysis

The results show that the proposed framework decreases the delay of task processing and task violation rate, and

the resource allocation process's running time also retains some consistency. The most important factor influencing the completion of user tasks is the computational capacity of the fog node. However, the competitiveness of multiple resources can affect the efficient distribution of resources in the fog environment due to the fog network's restricted hierarchy, network communication resources, and storage resources.

J. GENERALIZED MOBILE NETWORK ARCHITECTURE

1) Motivation

Serving the future obstacles and setting high capacity and low latency 5G networks are the main factors for transforming the mobile core network. In the existing literature, different technologies such as NFV and SDN are being discussed to meet the future needs of 5G networks. However, potential technologies such as the IoT, video networks, and others may have numerous requirements that emphasize the need for complex network features to be scalable.

2) Proposed Framework

The authors in [153] incorporate the principles of cloud computing, SDN, and NFV with mobile networks in the proposed architecture. The mobile network cloud includes mapping the network functions needed to integrate mobile networks with SDN technology in the proposed architecture. These functions are just controlled functions for the mobile network, i.e., MME, HSS, PCRF, and S/P-GW CPs. Transport, load balancing, defense, policy, charging, tracking, QoE, or resource optimization are additional functions. With this method, only strategically positioned SDN-capable switches and regular switches compose the user plane. SDN switches may either partially or fully replace the existing mobile transport network.

3) Critical Analysis

The proposed framework's testbed illustrates that some of the needs of 5G mobile networks are met by the planned architecture with SDN and NFV integration. The findings also show the advantages of SDN that enhance the successful and efficient use of resources with reduced overhead when used in the backhaul. The testbed results show high latency when transferring VMs with network components (e.g., MME or S/P-GW) due to HW failure or when additional processing resources are needed. The work lacks the discussion on virtualization, efficiency, and robustness of the proposed framework.

K. DISCUSSION OF RESULTS

NFV has been described as the most promising choice for the versatile programmability of network control functions and protocols for the dynamic use of network resources. SDN abstracts network resources into well-defined APIs, allowing IoT networks to be topology-independent. We thoroughly examined each of the primary studies chosen during the SLR and classified them into SDN-based NFV taxonomy based on the management challenges of IoT. With the support of SDN-based NFV solutions, we have addressed various techniques identified in the existing literature to solve IoT management challenges. Table-8 summarizes the merits and demerits of the SDN-based IoT-NFV solutions under consideration. Based on the above discussion, we summarize the answers to the research questions as follows:

RQ1: How can SDN-based frameworks provide efficient security solutions to manage IoT network-related security issues? To provide efficient security solutions to IoT networks with NFV based SDN integration, Alam et al. [47] explored how to incorporate the virtual security feature into the SDN-based NFV architecture. They are focusing more on network-layer security protocols such as routing algorithms, context-aware forwarding of IoT traffic etc. The authors introduced the NFV Management Module (NMM) and the safe Routing Management Module in [51]. The NMM includes a VNF container and a VNF manager to maintain the usable network service. The RMM node module securely stores and shares the energy-state information of its neighbours with the controller. The controller uses energy-state information to make the maximum number of NFV nodes possible and creates energy-aware routes at RMM. The authors in [146] propose a Docker-based framework for deploying virtual network security functions at IoT gateways. These virtual functions are stored in a cloud-based system. The IoT gateway is in charge of retrieving these cloud-based virtual network security functions according to their requirements. Authors in [152] proposed an AAA module in fog computing-based SDIoT architecture that provides an efficient security mechanism by intelligently controlling access to IoT devices through strict access and auditing policies. The majority of proposed SDN-based NFV solutions, according to data synthesis driven Table-7, lack security modules. Moreover, most of the proposed security modules are correlated to energy management solutions, and only a few of them are implemented in real-world scenarios.

RQ2: How can SDN-based frameworks provide effective fault tolerance management solutions to large-scale IoT networks? In [140], the authors explore fault tolerance techniques using NFV Management and Orchestration (NFV-MANO). The SDN controller manages the NFV orchestration unit responsible for providing the virtualized network's functionality through standard interfaces. After receiving the network topology and policy demands, the control module decides the optimal function assignments (assigning network functions to specific VMs). It converts the logic policy's specifications into optimized routing paths. The authors offered VIM (virtual infrastructure manager) in the proposed architecture to govern and manage Network Function Virtualized Infrastructure resources in its domain with fault management of hardware, software, and virtual resources in IoT networks in their paper [137]. The author [139] discusses the reference multi gateway architecture in which network elements such as the Network Control Centre (NCC) and the Network Management Centre (NMC) are

TABLE 7. Summary of SDN-based IoT-NFV solutions that addresses IoT management challenges

Exiting Work	Fault tolerance	Energy	Load balancing	Security	Scalability	Implementation
[20]					✓	
[30]				✓		
[47]	✓		✓		✓	
[51]		✓		✓		✓
[137]	✓		✓		✓	
[138]	✓		✓		✓	
[139]	✓		✓			
[140]	✓					
[141]	✓		✓			
[142]	✓		✓			
[143]					✓	
[145]		✓	✓		✓	
[146]	✓		✓	✓	✓	
[147]	✓			✓	✓	✓
[149]		✓	✓		✓	✓
[152]			✓	✓	✓	✓
[153]			✓		✓	
[154]			✓	✓	✓	

TABLE 8. Merits and demerits of proposed SDN-based IoT-NFV solutions

Existing work	Merits	Demerits
[51]	- The proposed architecture is implemented - Mainly focus on security feature only - Compared two architecture	- Work flow of the security feature is missing
[144]	- Decouple hardware from network services - Minimizing the cost of resource management - Discuss security management features	- Proposed architecture is generic - Architecture is not implemented - Does not provide architecture layer details
[145]	- Propose QoS framework for an IoT network - Discuss how SDN offers centralized control and view for different IoT services	- Architecture is not implemented - Discuss (QoS) in term of security only
[146]	- Focus on security and network performance with support of centralized management - SDN controller is distributed in nature	- Implementation and evaluation are missing - Proposed some security algorithm, but not discuss in detail
[149]	- The Proposed architecture is implemented - Focus only on energy management in IoT nodes - Result indicator shows better performance in energy consumption	- Architecture implementation based on small testbed - Focus only on energy management
[150]	- The proposed architecture is implemented - Proposed a method to defend IoT devices for different attacks through single VNF function	- Lacks discussion of the results in detail - No module is presented for resource, energy management
[151]	- The proposed architecture discuss NFV orchestration in detail	- Not implemented the proposed architecture - No module is presented for resource, energy management
[152]	- The proposed architecture discuss NFV orchestration in detail also Implemented	- No module is presented for resource, energy management
[153]	- The proposed architecture is implemented	- No module is presented for resource, energy management

responsible for managing fault tolerance performance by managing network function virtualization according to their needs. Authors in [142] discuss NFV-RA (network function virtualization-resource allocation)strategies with reference to QoS to manage fault tolerance. Table-7 shows that the majority of proposed SDN-based NFV solutions are proposed with a fault tolerance approach to manage IoT networks.

RQ3: What are the potential solutions regarding load balancing in SDN-based frameworks to manage IoT networks? Authors in [138] discusses the suitable approaches of load balancing in SDN-based NFV framework in controller with the help of access rules, such as Broadband Remote Access Serve (BRAS), etc. According to the authors in [152], a fog computing-based SDIoT architecture will effectively solve big data processing and network scalability issues in terms of load-balancing to manage IoT networks with the help of SDN based NFV framework. Table-7 shows that the majority of proposed SDN-based NFV solutions are proposed with a load-balancing approach to managing IoT networks, but a limited of them are implemented.

RQ4: What scalable solutions can be offered by SDN-based frameworks to manage IoT networks? With the support of a distributed SDN controller, Li et al. [155] proposed an SDN-based NFV architecture to manage IoT networks through virtual networking features such as routing, safe tunneling between IoT gateways, and traffic prioritization for QoS in a scalable manner. The authors in [146] suggested a Dockers-based architecture for deploying virtual network

security functions at IoT gateways. These virtual functions are stored in the scalable cloud. The IoT gateway is in charge of retrieving these virtual network security functions from the cloud. Afek et al. [149] proposed a scalable SDN-based NFV architecture for the various distributed scalable forms of attacks, such as DDoS, etc. According to Table-7, most proposed SDN-based NFV solutions are scalable, and hence this challenge is tackled on different levels with various solutions.

RQ5: How can SDN-based frameworks enable efficient power consumption in IoT networks? In [145], the authors presented a distributed, secure Black SDNIoT architecture for smart cities that included NFV implementation. For energy conservation, load balancing, and network scalability, the proposed unified Black SDN-IoT architecture with NFV is being considered for smart cities. The authors implemented several hierarchical SDN controllers in the proposed system to improve availability, credibility, and confidentiality, among other things. Li et al. [155] in proposed architecture have energy efficient secure networking features at network layer such as routing, safe tunneling between IoT gateways. According to Table-7, the majority of proposed SDN-based NFV solutions are missing energy management solutions. Most of the proposed efficient energy management modules are related to security management solutions, and very few are implemented.

VI. MIDDLEWARE-BASED SDN MANAGEMENT FRAMEWORKS

A middleware layer for the IoT environment is required for different applications. The common goal of all the middleware layer development in IoT is to develop a framework that can allow a plug-n-play adaptation layer [156]. Among all the various devices belonging to diverse IoT domains, it is difficult to define and enforce a common standard. Middleware acts as a bond that joins together the heterogeneous components [157]. IoT has a software framework known as middleware that basically provides an abstraction from items to applications and offers multiple services. The middleware layer addresses interoperability across heterogeneous devices that serve in various application domains, adaptations, context awareness, discovery and management of devices, scalability, privacy, and security in the IoT environment [158].

SDN-based middleware is now becoming quite popular as a way to manage and control networks. A typical SDN-based middleware is shown in Fig. 23. In this architecture, middleware logic connected with the software components resides at the SDN-based CP. Switches send the OF messages to the middleware, which is processed by the middleware components [159]. These components perform the task of adapting the network behavior and sending messages back to the network devices. Such architecture enables adaptation of the network based on the prevailing network situation controller for classifying the legitimate user that has been involved in the network [160]. This section will discuss all the efforts that have adopted a middleware-based approach to manage IoT networks.

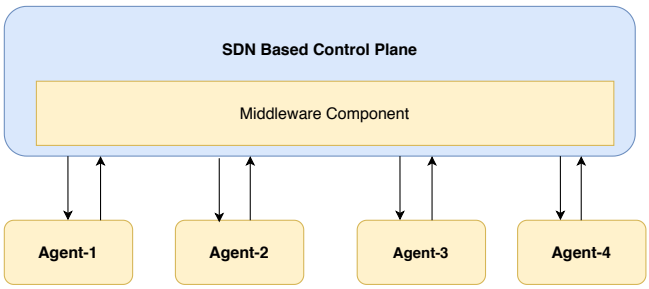

FIGURE 23. Middleware-Based SDN Controller.

A. CLOUD-BASED PUBLISH/SUBSCRIBE MECHANISM FOR IOT

1) Motivation

Mobile crowdsensing (MCS) is a new trend of development in IoT applications. Mobile nodes are scattered in large network forms capable of sensing and computing collectively share data with the help of distributed cloud structure. Due to mobility nature architecture, MCS has to face dynamic environments comprising sensors, heterogeneous mobile devices that make it necessary to have energy efficiency and context-aware for community sense. That means both the sensing and data transmission processes from mobile devices and the cloud need to be managed effectively.

2) Proposed Framework

Antonic et al. [161] proposed a solution (CUPUS) based on content-based publish-subscribe with distributed cloud help, as shown in Fig. 24. The center cloud nodes are responsible for collecting information from the subscriber nodes for processing and data analytics. The proposed framework consists of two essential components, the mobile broker and the cloud broker. The mobile broker module is responsible for data filtering and data acquisition of connected sensors., while the cloud broker module in the proposed framework is responsible for processing a big data stream to perform data analytics for the IoT gadgets. The authors have evaluated their proposed framework in terms of propagation delay from IoT nodes to central cloud nodes. The authors used a Citrix XenServer virtualization software and 20,000 subscription nodes in a simulated environment to implement the proposed framework.

3) Critical Analysis

CUPUS middleware is designed for handling the resource-constrained requirements of IoT gadgets. The result suggests that the proposed framework controls the data density by filtering closer to the production place. Authors never discuss cloud brokers' and mobile brokers' details in the implementation, which means that the result can be deflected in different scenarios.

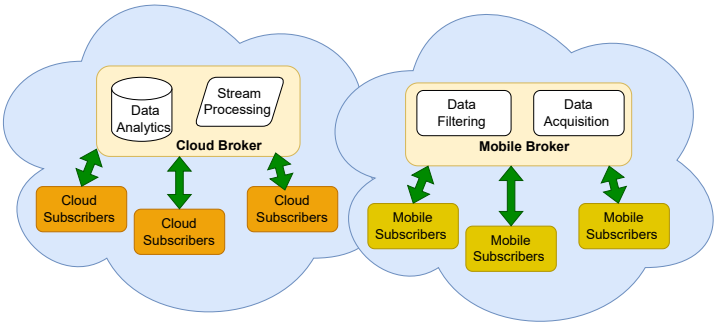

FIGURE 24. Cloud-based publish and subscribe middleware for IoT.

B. PUBLISH/SUBSCRIBE SYSTEM FOR LOAD BALANCED TOPIC-BASED SDN

1) Motivation

IoT in the future has severe challenges due to the massive stream of data movement in multi-source sensors. The traditional techniques in IoT infrastructure to address these challenges are insufficient because of the absence of a global traffic information center. The topic-based publish-subscribe system is a special kind of publish-subscribe mechanism in which events are published with specific identifiers called topics. Publisher broadcast this topic to the concerned subscribers. This publish-subscribe mechanism tends to manage the IoT network more efficiently with the help of SDN.

2) Proposed Framework

Wang et al. [162] have proposed an SDN topic-based publish-subscribe system known as SDNPS. The proposed architecture is partitioned into multiple clusters. These clusters are belonging according to their regional characteristics. The different logically autonomous areas represent each cluster in the topology. The clusters are communicating with each other through the border gateway. At the top of the proposed architecture, global servers manage the whole topology and compute routing efficiently. The authors proposed a framework to implement an efficient routing protocol based on topic connected overlay called the minimal cost topic-connected overlay (MCTCO) that operates by creating an improved routing plan. The global view of the topology is acquired by collecting a link-state. Publish/subscribe paradigm then guarantees that distributed every new event to the connected subscribers whose interest in the topic similar. The proposed framework consists of a three-layer, switch hardware layer, cluster controller layer, and the global management layer. The switch layer is responsible for taking information from the IoT nodes or agents with the OF protocol's help or Southbound API and pass on to the SDN controller. The global management layer consists of two types of servers for single-point failure, i.e., a major server and a standby server. These servers contain information on overall topology and routing policies. Moreover, the proposed framework maintains two kinds of topology, i.e., subscription topology and physical topology. IBM server with 16 GB memory is used to implement the scenario in three-hop topology.

3) Critical Analysis

One of the shortcomings of the proposed framework is that SDNPS has to compute the topic tree and maintain clusters in the network. The creation of a topic tree would require extra computation. The authors never discuss how to manage the cluster of SDN controllers because cluster management would require additional computation and storage to preserve the cluster state.

C. PUBLISH/SUBSCRIBE ENABLED SDN FOR IOT

1) Motivation

IoT infrastructure in the future will face many challenges related to mobility management, integration with traditional communication protocol, security, etc. There is a high need for such a framework that overcome these issues and provide improved service in the IoT network. SDN has the potential to provide such a novel IoT framework with the help of data distribution service middleware.

2) Proposed Framework

Hakiri et al. [163] identifies five main barriers to networking. Current standardization attempts at various levels of the protocol stacks for IoT are isolated. The authors have proposed 6LowPAN protocols on the network layer between MediaAccess Control (MAC) and IPv6. The protocol requires IPv6 to run on resource-limited computers. ROLL (Routing for Low Power and Loss Networks) often addresses routing problems for low power applications. CoAP (Constraint Application Protocol) is on the application layer, a specially developed application protocol for resource-limited devices that comply with the 6LowPAN protocol to provide application services. M2M movements have arisen to promote the implementation of the end-to-end IoT architecture. These standards need to be combined and interoperated to make them the possible end of the IoT end architecture. Since IoT devices are highly mobile, the need to handle the versatility of IoT devices has earned a high degree of interest from them effectively. SDN can help manage versatility as it maintains a full view of the network but offers an increasingly mobile network which will be a challenge. In IoT environments, middleware is required to propagate activities to destinations of interest in an asynchronous position. TCP is not sufficient for IoT scenarios, and a reliable transport protocol is expected to be studied for IoT situations. Finally, there is no infrastructure to provide defense in IoT as traditional defense systems are dynamic and complex. Hakiri et al. [163] added Data Distribution Service (DDS) middleware between IoT app and SDN (Open daylight) controller with the help of the NBI interface. The proposed framework addresses the highlighted issue as follows. DDS middleware allows cross-domain or cross-platform interoperability, allows reconfiguration of IoT devices according to their environment requirement, supports multiple communication patterns in a large distributed IoT

system, and provides security mechanism with the help of imposing security policies.

3) Critical Analysis

The proposed architecture is merely a conceptual representation, and the authors have provided no implementation of their approach. Thus, there have been no reliability tests and evaluations of the architecture under different production conditions. This middleware approach, together with the SDN controller, often adds extra strain or overheats to the provision of IoT applications.

D. PUBLISHED/SUBSCRIBE ENABLED COMMUNICATION PLATFORM FOR IOT USING SDN

1) Motivation

The exponential growth of IoT gadgets and services/applications opens new challenges for researchers in managing IoT services/applications efficiently. IoT applications need QoS requirements, such as no latency and high data rate required in real-time data analytics and processing. Differentiated QoS is another critical issue that plays a vital role in creating serious delays in the IoT network.

2) Proposed Framework

The authors in [164] proposed an SDN-based publish/subscribe communication platform responsible for countering typical IoT networks' issues. The author implements a topic-based publish/subscribe paradigm under SDN as a data distribution service, which communicates events between IoT connected nodes. The proposed architecture consists of three layers: the infrastructure layer, the network layer, and the application layer, as shown in Fig. 25. The infrastructure layer consists of sensors/actuators responsible for generating data and then delivering it to the connected nodes. The network layer consists of many SDN-configurable switches responsible for providing network service. SDN controller is responsible for managing sub-modules such as topology management, routing service, flow-table management, packet scheduler, etc. The application layer interacts with the message bus called local processing brokers, which implement the topic-oriented publish/subscribe service. The message bus receives data from sensors/actuators, combines them in a predefined format, and puts them on SDN infrastructure to transmit. The network layer is responsible for forwarding events efficiently and providing differentiated services to meet the event constraints such as end-to-end latency, loss rate, etc.

3) Critical Analysis

The proposed architecture facilitates the access of various IoT services to a single middleware approach. The author implemented a prototype by considering the same deployed topology in District Heating Control and Information Service System (DHCISS) in Beijing and evaluating the proposed architecture's correctness and feasibility. The author does not discuss the detail of the performance evaluation parameter involved in the proposed architecture's throughput. The presented evaluation graph does not contain sufficient information; no security scenarios are discussed in the proposed system.

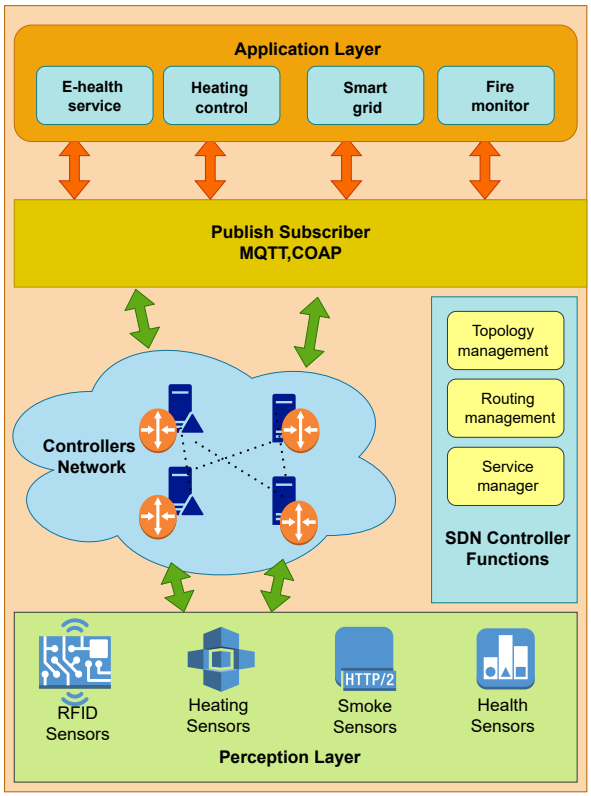

FIGURE 25. SDN-based publish/subscribe service framework.

E. SDN-INTEGRATED FRAMEWORK FOR IOT TRAFFIC MANAGEMENT

1) Motivation

The increasing usage of IoT raises challenges in managing heavy network traffic and maintaining the quality of service requirements. Most IoT devices have differences in processing, storage, power, and functionality, which cause complex issues for QoS, resource allocation, and network configuration in the IoT network. There is a high need to build middleware-based QoS strategies to better serve IoT applications by knowing how IoT devices transmit data and how applications consume that data.

2) Proposed Framework

The authors in [165] proposed an SDN QoS control-based publish-subscribe model to manage the IoT networks. The PS-IoT SDN framework is a QoS-aware framework for managing IoT traffic aggregated into Fog-like IoT gateways along the network edge. The author first discusses the existing architecture PSIoT-Orch framework created to manage IoT networks during massive traffic situations generated by growing

IoT devices. The architecture uses Publish/Subscribe to allow IoT data transfer among producers and consumers nodes and efficiently handle network resources at the edge level based on the QoS requirement. A traffic orchestrator module in the proposed architecture responsible for managing traffic policies in IoT networks, IoT gateway, or data aggregators (IoT gateway) acts as Pub/Sub. The Pub/Sub component is accountable for managing data-prepossessing, backup, or caching and cloud processing center. Here the centralized orchestrator must play an essential role in the communication of clients and producers nodes. The centralized orchestrator is responsible for the flowing of IoT data according to the IoT data characteristics. The orchestrator knows each IoT gateway according to the topic subscription. The objective is to accomplish both edge level and system level QoS in the IoT network by utilizing the QoS management capabilities coupled with SDN-based network link bandwidth allocation. The PS-IoT orchestrator communicates to the SDN controller by requesting the communication path set up with the interface provided in the SDN controller sub-block. The MAM module is responsible for bandwidth sharing through the bandwidth allocation model strategy. This module keeps track of the used bandwidth for all links over the path between IoT producer and IoT consumer calculated using the routing algorithm. The SDN controller creates entries on the Open Flow switches involved in the path.

3) Critical Analysis

The proposed architecture focuses on how data is transferred, discovered, shared, and consumed to manage IoT networks better to adopt the SDN paradigm. The proposed architecture's evaluation results generate more massive throughput when bandwidth is distributed among the framework's IoT QoS levels. With these positive results, the proposed framework is validated for its usefulness in managing QoS for IoT traffic. The authors only focus on QoS in the implementation, and the security and scalability issues are not discussed in the implemented scenarios.

F. MIDDLEWARES SD-IOT FRAMEWORK

1) Motivation

Conventional storage and security mechanisms cannot be implemented to manage the IoT devices and networks due to their limited resources, so there is a need for such a platform to overcome this problem in IoT networks. With NFV middleware, multiple SDN-based functions are implemented to manage the IoT networks.

2) Proposed Framework

Authors in [166] proposed an architecture that has three components, i.e., physical layer, control (middleware) layer, and application layer. IoT sensors are responsible for collecting data at the physical layer and providing this data to the successive layer. The physical layer maintains the database pool for different reasons, such as keeping the configuration of each IoT connected node. Physical Layer communicates with middleware with the help of South-bound API in the SDN controller. The middleware comprises different network functions based on the SDN controller, and the lower layer calls the IoT controller according to their requirement.

3) Critical Analysis

The author's proposed framework is very generic. Integration of various SDN-based network functions such as Software defined storage (SDStore) and Software defined security (SDSec) should be evaluated because such functions will produce overhead in the IoT traffic. No implementation and evaluation results of the proposed architecture are presented; hence there is no way of knowing whether such middleware is feasible to implement.

G. STATEFUL SDN SOLUTION FOR WIRELESS SENSOR NETWORKS

1) Motivation

Wireless sensor networks have similar challenges that IoT networks face, with limited energy, processing, and memory availability. There is a need for middleware solutions that can manage the wireless network more efficiently and overcome the existing problems in the WSN domain. With the help of the SDN-WISE solution, wireless networks can be efficiently managed and became adaptable with programmability.

2) Proposed Framework

Authors in [167] proposed a framework based on SDN. SDN-WISE network maintains three data structures, state array, IDs array, and WISE flow table. ID array is responsible for keeping sensor IDs in a current scenario. State array maintains the table of physical states and statistical reports of existing IoT nodes, while the Wise flow table is accountable for taking information from the controller and build the flow table as per the requirement. The controller is responsible for defining the network management policies to the connected IoT nodes. The sensor nodes work under the DP protocol stack to communicate with other sensing nodes. The sink node provides a bridge between sensors and controllers through WISE-Visor. This middleware is responsible for generating local topology information with the help of the topology discover protocol. The sensor nodes at the forwarding layer are accountable for handling the sensor traffic according to the flow table. At the INPP layer, data aggregation is performed. The TD (Topology discovery layer) is responsible for encapsulating the information of topology in the header.

3) Critical Analysis

The proposed framework is implemented with the help of wireless module EMB-Z2530P, which acts as sensor nodes. The testbed is created with the help of five sensor nodes and one sink node. SDN-WISE Controller using Dijkstra's algorithm for routing the packets in which 5000 data packets of connected sensor nodes are sent every 15 seconds.

The evaluation results show that the proposed framework is efficient under a particular testbed and allows adaptability according to the requirement. The authors of the proposed framework never discuss the overhead of topology discovery protocol present in WISE-VISOR middleware.

H. SOFTWARE-DENED NETWORKING PRINCIPLES IN WSN

1) Motivation

SDN is an essential building block for the structured low-cost application hardware off-the-shelf and still achieves customization necessary for individual deployments. SDN can be used for various purposes, including networking, network processing, and WSN administration tasks.

2) Proposed Framework

Jacobsson et al. [168] proposed an SDN-based WSN architecture to manage WSN with SDN layers. The proposed architecture counters the issue of scalability and reconfiguration of WSN networking. The WSN node is attached to a local controller that accepts and executes directions from the central controller. At the top of the controller, one or more applications can be placed. Local controllers are present within the sensor nodes that will change both the MAC and the routing behavior of the sensing nodes themselves. These controllers take commands from the central controller. The local controller that resides in the sensor nodes is responsible for modifying and controlling the code. Modification can be achieved either by altering the parameters (e.g., adjusting the central frequency of the antenna, MAC layer repropagation cap, modifying the outputs in the forwarding table, etc.) or by installing new features (e.g., virtual machines, native software, and dynamically connected libraries) that would change the behavior of the network. Forwarding and many routing decisions are made at individual nodes. However, long-term decisions, such as the protocols and parameters to be used, are taken by the central controller. The central controller is responsible for discovering the current topology and connection performance of the connected nodes. To determine the quality of the connections, the controller used Link Quality Estimation(LQE). The controller is responsible for predicting the WSN node's behavior and network lifetime and performance.

3) Critical Analysis

The authors have discussed the work as a conceptual exercise and have not presented any prototype implementation and evaluations of different scenarios. Even within the conceptual framework, it is unclear how the central and local controllers would synchronize and coordinate with each other or how the management functions are distributed between these two types of controllers.

I. SDN APPROACH TO IOT NETWORKING

1) Motivation

The IoT is projected to contain billions of connected devices, rendering the provision and operation of certain IoT networking services more difficult. Indeed, IoT services are somewhat different from legacy Internet services because of their dimensioning statistics and because IoT services vary drastically in design and constraints. For example, IoT services also rely on energy and CPU-like sensor technologies, regardless of whether the use is for home automation, smart building, e-related health, or regional or national power or water metering. Some IoT services, such as dynamic monitoring of biometric data, exploitation of confidential information, and privacy, need to be safeguarded whenever this information is transmitted over the IoT network infrastructure. Authors in [169] explores how SDN can enable the deployment and operation of certain advanced IoT services, regardless of their existence or scope.

2) Proposed Framework

Jacquenet et al. [169] proposed an architecture for SDN-based IoT networks. In addition to the proposal, the authors also introduced two IoT services: eHealth and energy management. eHealth requires network infrastructure which is highly reliable in preserving data integrity. In addition to this, some eHealth scenarios require quick reaction time and would probably need dynamic route computation for sending data. The authors' second use case is large-scale IoT dynamic energy distribution management. With an SDN-based IoT network for energy distribution, it will be possible to effectively implement traffic forwarding policy in the IoT network. Through the help of data analytics on the data collected from the IoT network events, the performance of the IoT infrastructure is evaluated. The research has effectively used SDN to manage the IoT services. They have distinguished policies for traffic forwarding to prioritize traffic in the IoT network.

3) Critical Analysis

The authors do not discuss the prototype implementation of the proposed framework. Details of the algorithms used for architecture implementation were also found to be missing. The work is simply a conceptual undertaking in its current state. Details of the functions of the proposed architecture are not provided even within the conceptual model. The authors claim that most IoT gateway and node features are relocated to the IoT system and virtualized as the VNF, coordinated by the IoT network by the SDN/NFV controller or orchestrate. This proposed virtualization over IoT nodes is not feasible due to the IoT device's resource constraint nature.

J. MOBILITY MANAGEMENT IN URBAN-SCALE SDN-BASED IOT

1) Motivation

IoT flows are distributed in nature and need to be regulated. The IoT controls and commands are grouped into the geographical regions within the IoT networks. An interactive

view can be used in multi-network flow to select better access points. A special overlay architecture that can primarily contribute to stability mitigation and fault tolerance in SDNIoT will be highly suitable. In a single share, IoT gadgets may connect various types of local switch-related access points.

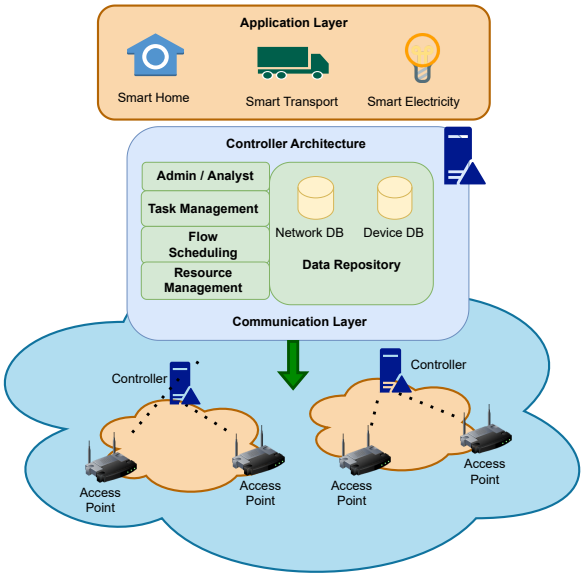

FIGURE 26. UbiFlow system architecture.

2) Proposed Framework

Wu et al. [170] have proposed Ubiflow, an SDNIoT architecture, as shown in Fig. 26. Ubiflow has multiple controllers. The geographical regions within the IoT networks are split between these controllers, resulting in distributed control of IoT flows. Ubiflow controllers schedule the flows according to device requirements and offer a unique overlay structure achieving mobility management and fault tolerance in SDNIoT. The core components of the system architecture of Ubiflow are switches, access points, data servers, controllers, and Internet devices. The data collection component gathers network/device information from IoT multi-network neighborhoods and caches it in a database. Layered components use gathered data in the controller. The component responsible for task resource matching matches task requests with existing resources in multi-network.

3) Critical Analysis

The framework of Ubiflow solves IoT control problems, such as error sensitivity and load balancing. Several controllers were implemented in the architecture that can create issues related to synchronization, but the authors never addressed these issues. The proposed framework is implemented with the help of Omnet++. ORBIT is used as a wireless testbed for experiments to evaluate the proposed architecture where performance and time are observed. It consists of 400 radio nodes. ORBIT has an open-light controller for WiFi and WiMAX. The framework's scheduling algorithms are compared with the conventional famous scheduling algorithms of Devoflow and Hedera.

K. DESIGN OF LR-WPAN IOT SYSTEMS WITH SDN

1) Motivation

Despite the current developments in WSN and IoT, the existing Internet architecture can not meet the high volume of new traffic trends from smart sensing systems. SDN has emerged as a smart solution to improve network programmability, agility, versatility.

2) Proposed Framework

Hakiri et al. [171] proposed the SDN-based framework for sensors, which represents a new SDN framework that meets various special WSN requirements. The proposed framework consists of two planes, CP and the DP. The interaction of the CP and the DP takes place with the help of the OF protocol. The CP comprises multiple functions and is responsible for providing topology discovery, mobility, and managing network policies to the DP. The DP, which is also called the sink, is accountable for performing packet engineering and packet aggregation. TDMA layer is responsible for providing dynamic and flexible data forwarding to the physical layer.

3) Critical Analysis

The work lacks evaluations of the performance of the architecture under different scenarios. Due to this reason, it is not possible to know the overhead of provisioning services by the controller and the proposed TDMA protocol. It is also unclear the sequence of messages exchanged by the CP and the DP for provisioning topology discovery and virtualization service over the network. Detailed evaluations should also be done to figure out the use cases under which the WSN traffic load and the proposed programmable layer would be cost-efficient in resource consumption and otherwise.

L. SDN FOR INDUSTRIAL IOT

1) Motivation

Industrial Internet of Things (IIoT) is a new subfield of IoT that deals with deploying a wide range of sensors to track supply chain, manufacturing, and other industries in real-time. IIoT deployments need to address information-based interactions whereby the system's experiences change over time, depending on the given knowledge.

2) Proposed Framework

In Industrial IoT, application needs vary from near real-time data access to asynchronous data access depending upon certain triggered events. Furthermore, the lack of technology standardization is a significant impediment to the adoption of Industrial IoT solutions. Due to the lack of standards, interoperability between different systems and technologies has become a real pain point. This issue can be solved by standardization of interface intercommunication among varying components developed by various vendors. The proposed

architecture of Wan et al. [172] as shown in Fig. 27, provides information collection, data transmission, and processing services. The data transmission system passes detected data to the commercial cloud from the network.

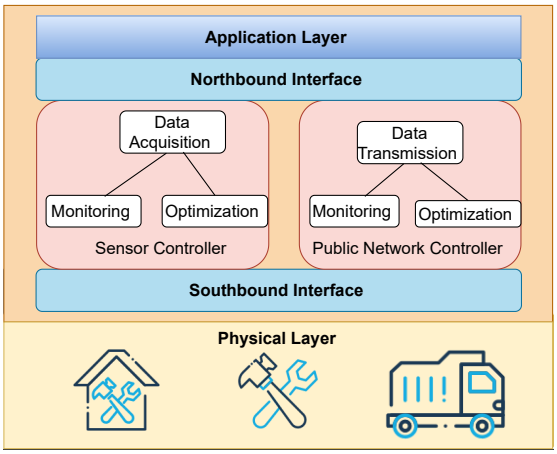

FIGURE 27. SDN for Industrial IoT.

3) Critical Analysis

The authors developed a model consisting of a cloud data center, an industrial machine with AVG, IWN, RFID scanner, conveyor, etc., to analyze their system. The proposed framework is contrasted with the SDNIIoT architecture. For the planned structure and traditional systems, energy efficiency and usage are analyzed. Results indicated that the IIoT SDN-based design requires less power, is stable, and facilitates autonomous industrial decision-making.

M. SDN-BASED FRAMEWORK FOR WIRELESS SENSOR NETWORKS

1) Motivation

The motivation is to have effective control of the communication infrastructure, reduce the processing load of the forwarding nodes, increase the network's reliability, and reduce the energy consumption within the WSN and IoT.

2) Proposed Framework

Zhou et al. [173] have proposed an SDN framework for Wireless Sensor and Actuator Networks (WSAN). As shown in Fig. 28, the WSAN structure consists of three different layers: Application, CP, and DP. The conventional WSAN protocol stack has a shared plane that communicates with five levels of protocols (application, storage, network, medium, and physical access) to decide. Rather than SDN-based, WSAN operators will make routing decisions. The CP module is composed of with SDN controller and a scheduling engine. Due to this reason, SDN-based CP can make decisions with regards to commands from devices in a more efficient and robust manner.

3) Critical Analysis

Zhou et al. [173] framework is quite effective in managing mobility and ensuring energy conservation within the WSAN. However, there is no discussion on how security and fault tolerance would be handled. Furthermore, the proposal necessitates significant changes to the protocol stack of WSAN to adopt SDN into the WSN stack. Further studies regarding load management and balancing should also be done to find out how effectively the controller manages data load from the DP and responds to requests from the applications plane.

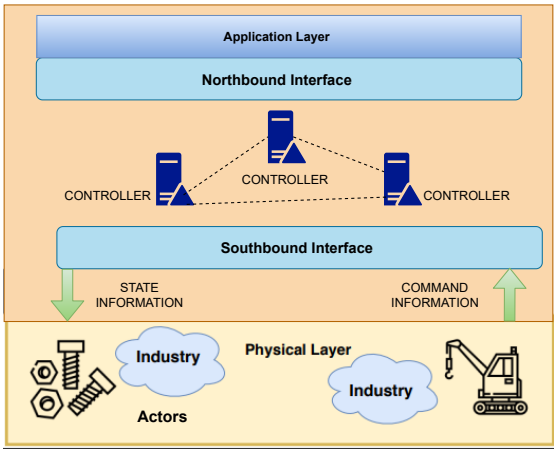

FIGURE 28. A WSAN framework based on SDN application.

N. SDN-BASED REFACTORED MIDDLEWARE FOR IOT

1) Motivation

SDN allows for a redesign of the middleware architecture to improve service interconnection, management, and the deployment of new monitoring scenarios. The middleware can also be refactored to accommodate a variety of services. The motivation behind the author's proposed framework [174] is to solve common difficulties in IoT contexts, focusing on connectivity, security and privacy, management, and data structure, particularly in health monitoring scenarios.

2) Proposed framework

Ariza et al. [174] expanded its support towards SDN technologies and proposed the REMOA middleware framework. Complex networks across access points (AP), which are spread across more machine-driven databases, are introduced in the architecture. The AP'S and network servers fulfill the function of the actual proxy unit. Flow-based APs is transmitting the packet. Things collected data is sent to services via the IPSec tunnel. The handling of objects formerly focused on SNMP is now centered on OF counters. The ThingsFlow application is available through APs and provides a timestamp that shows when the counter is being found. Counters are saved in ThingsFlow and retrieved through services that implement control mechanisms. The gateway passes access points packets (AP) in compliance with OF rules. With

the addition of SDN, middleware capabilities have been expanded, and every AP can now provide several services.

3) Critical Analysis

The authors have not presented any evaluations, so it is unclear how much additional overhead would be caused by message passing between the different modules after the refactoring of REMOA gateway. Studies are also needed to find out the degree of complexity that has been added as a result of incorporating new modules in the REMOA architecture.

O. ENHANCING MIDDLEWARE-BASED IOT APPLICATIONS

1) Motivation

With the arrival of the new paradigm such as NFV, it is now possible to deploy any network function such as switching, traffic monitoring, load balancer, etc., offering the required functional capabilities in a virtual form rather than implemented on dedicated equipment.

2) Proposed Framework

Authors in [175] proposed a middleware framework based on a self-adaptation of QoS oriented mechanism, consisting of autonomic computing (AC) model to interact with sensors and effectors in the IoT. Sensors are basically monitoring the service requirement, and the effector is implemented the required QoS to the connecting nodes with the help of the middleware MW entity (Public cloud). This MW entity implements a QoS microservice as a virtual network function. The autonomic computing model is responsible for monitoring the system with the help of sensors and reconfiguring the system according to the requirement, and finally executing the plan with effectors' help.

3) Critical Analysis

The authors implement different algorithms at the application network function named redirector, compressor, and decompressor in a transportation use case. The result clearly shows the better response time at adaptation QoS with no adaptation. The proposed framework has only focused on a specific use case of transportation of QoS self-adaptation. The authors never discussed the detailed implementation of connected actors in the proposed architecture and nor mentioned how to handle the security risk of the middleware data center that provides the QoS service in the form of a virtual network function. The implemented algorithm's performance is a question mark because the algorithm was tested only on the specific use case of transportation.

P. SDN FRAMEWORK FOR INDUSTRIAL IOT

1) Motivation

The IoT-based smart industry aims to manage the industrial process to achieve better performance in the industrial revolution; however, challenges are raised due to the massive IoT gadget deployment in smart industries. The authors in [176] tries to meet these challenges by presenting an SDN-based solution on OpenDaylight (ODL) controller to manage the industrial IoT scenario.

2) Proposed Framework

The IIoT domain consists of IoT devices from different communication standards such as sensor motes, RFID, BLE working under ProfNet, Ethercat, CAN Bus, and Modbus. The IIoT network also contains the conventional IT enterprise network, composed of routers, switches, PCs, printers, etc. The proposed architecture [176] has two ODL SDN controllers for managing the industrial process. The framework is for multiple purposes such as backup, maintaining fault tolerance, managing security risk, etc. The authors mentioned that the number of controllers in the proposed architecture could vary according to the IIoT domain's situation. ODL uses a software functionality called Virtual Tenant Network (VTN) to control controllers' cluster. The authors define two types of SDN controllers in the proposed framework. The first controller is for the IT network's control, and another controller is for the IoT network. The industrial machinery networks consist of devices running on different communication protocols such as Modbus, CAN Bus, and Ethercat. These protocols cannot communicate directly with the IoT domain due to their data format and communication protocol's incompatibility. To communicate with the IoT domain, these devices use a middleware approach based on OPC UA client-server architecture that communicates with the controller.

3) Critical Analysis

The proposed architecture is a conceptual solution. The authors did not implement their approach because there were no performance studies and evaluations of the architecture under various production scenarios. This proposed SDN controller deployment solution has some advantages and disadvantages. The benefits are modularity is that it provides more efficient management of applications. The disadvantage is hardware needs, such as high-powerful computers, allocating for each controller, and assigning backup for controllers.

Q. DISCUSSION OF RESULTS

IoT applications involve a variety of layers having different processes, hence managing such IoT networks necessitated the use of an abstraction/adaptation layer. Middleware hides all the complexities of diversity by providing API for physical layer communications and other required services to applications. To overcome the management issues of IoT networks, SDN-based Middleware acts as a link connecting heterogeneous components. Based on the management issues of IoT, we thoroughly investigated each of the major studies chosen during the SLR and classified them into SDN-based middleware taxonomy. We examined several strategies described in the existing literature to overcome IoT management difficulties using SDN-based middleware solu-

TABLE 9. Summary of SDN-based Middleware Solutions that addresses IoT management challenges

Exiting Work	Fault tolerance	Energy	Load balancing	Security	Scalability	Implementation
[155]					✓	✓
[157]	✓		✓		✓	✓
[158]	✓		✓		✓	✓
[159]	✓		✓		✓	✓
[160]	✓		✓	✓		
[161]	✓		✓			✓
[162]	✓					✓
[163]	✓		✓		✓	
[164]				✓		
[165]	✓		✓	✓	✓	✓
[166]	✓		✓	✓	✓	
[167]		✓	✓		✓	
[168]	✓	✓			✓	
[169]		✓		✓		
[170]		✓	✓			✓
[171]		✓	✓			✓
[172]	✓	✓	✓		✓	✓
[173]					✓	✓

tions. Table-10 summarizes the merits and demerits of the SDN-based Middleware solutions for IoTs that were selected through the SLR. Based on the foregoing discussion, the answers to the research questions are presented as follows.

RQ1: How can SDN-based frameworks provide efficient security solutions to manage IoT network-related security issues? As mentioned before, conventional security mechanisms cannot be implemented to manage IoT devices due to their limited resources. Authors in [162] highlighted the need for such a middleware-based approach to address IoT management challenges; therefore, they proposed a middleware-based SDN solution to manage the IoT networks. They proposed various virtual network security functions at the controller layer, such as Software-defined storage (SDStore) and Software-defined security (SDS). Authors in [157], [159] discuss the general way to implement the middleware SDN-based solution in order to maintain efficient security in IoT networks discuss the security in terms of application layer protocol, network layer protocols such as CoAp, MQTT, HTTPS, IPSEC etc.Table-9 clearly shows significantly fewer efforts are made to address the security challenges in IoT networks with the help of a middleware-based SDN framework.

RQ2: How can SDN-based frameworks provide effective fault tolerance management solutions to large-scale IoT networks? Authors in [168] proposed a framework based on SDN with scalability and reconfiguration features to address fault tolerance in WSN. The IoT nodes are connected to a local controller, which receives and processes commands from the central controller. One or more applications can be placed at the top of the controller with the help of an efficient load balancing approach. Local controllers are located within sensor nodes, and they can affect the MAC and routing behaviour of the sensing nodes. The central controller issues command to these controllers. The code is modified and controlled by the local controller, which is located in the sensor nodes. The Industrial Internet of Things (IIoT) is a new area of the Internet of Things that uses various sensors to follow supply chains, manufacturing, and other industries in real-time. IIoT installations must address information-based interactions, in which the system's experiences change over time as a result of the knowledge available. The authors' in [172] proposed framework results indicated that the IIoT SDN-based proposed framework requires less power, is stable, and facilitates autonomous industrial decision-making with the help efficient fault tolerance scheme. They also highlighted the importance of fault tolerance solutions in IoT networks.

RQ3: What are the potential solutions regarding load balancing in SDN-based frameworks to manage IoT networks? The increasing usage of IoT raises challenges in managing heavy network traffic and maintaining service requirements. Most IoT devices have differences in processing, storage, power, and functionality, which cause complex issues for QoS, resource allocation, and network configuration in the IoT network in terms of load balancing. The authors in [165] proposed an IoT network management paradigm based on SDN QoS control and publish-subscribe. The proposed framework is a QoS-aware framework for managing IoT traffic aggregated into Fog-like IoT gateways along the network edge with the help of an efficient load balancing mechanism. The authors highlighted some critical parameters to address IoT networks in load balancing, such as QoS, network configuration, etc. The massive stream of data transfer in IoT networks poses severe issues in the future. Wang et al. [162] proposed SDNPS, a topic-based publish-subscribe system based on SDN. The architecture is divided into numerous clusters. These clusters are grouped based on their regional characteristics. Several conceptually autonomous areas represent each cluster in the topology. Through the border gateway, the clusters communicate with one another. They proposed the minimal cost topic-connected overlay (MCTCO), an efficient routing protocol based on topic connected overlay that operates by generating an optimum routing schema based on load balancing techniques. Authors in [160] discuss the concept of adaptive load balancing technique with the help of detecting

TABLE 10. Merits and demerits of SDN Middleware based Solutions for IoT Networks

Existing work	Merits	Demerits
[155]	- The proposed architecture is implemented - Proposed IoT service deploy quickly	- Discussion of operational component is missing - Security management is the future task
[161]	- The proposed architecture is implemented - Focus on load balancing in IoT nodes - Result suggested optimal load distribution in IoT nodes	- Focus only on the load balancing feature - Limited testbed for implementation - Other modules such as security, energy management are missing
[162]	- The proposed architecture is implemented - The SDN controller is implemented in a distributed manner - Framework manage two kind of topology that is subscription and physical	- Creation of topic tree created an extra burden to the framework - No discussion on how to manage SDN controller cluster
[163]	- Proposed DDS middleware enable cross-domain interoperability - Presented load balancing and security features - Discussed reconfigure of IoT devices according to their environment	- The architecture is not implemented
[164]	- Proposed architecture facilitate the access of various IoT services to a single middleware approach - The proposed architecture is implemented - Mainly focus on load balancing	- Result does not contain much information - No security scenario is proposed
[165]	- The proposed framework is implemented - Result generate massive throughputs	- Focus only on QoS in implementation - Other features such as security and resource management are missing
[166]	- Discussed middleware that comprises of different network functions based on SDN controller according to their requirement	- No implementation of the proposed framework - Integration of SDN-based network function is missing
[167]	- Framework that maintains data structures, state array, ID array and Wise Flow tables - Controller is responsible to defined network policies - The framework is implemented	- SDN controller uses only Dijkstra algorithm - No discussion on the overhead of topology discovery protocol
[168]	- Discuss how SDN-based architecture is used to manage WSN - Proposed Link Quality Estimation algorithm for quality connection	- The proposed framework is not implemented
[169]	- Discuss two real usecase of IoT networks that is e-Health and energy management	- The proposed framework is not implemented - Detailed discussion of proposed algorithm is missing
[170]	- The proposed architecture is implemented - Slice network services according to their respective controllers - Focused on load balancing	- Several controllers are implemented that can create the synchronization problem - Limited testbed for implementation
[171]	- The proposed architecture is implemented - Proposed SDN-based model that fulfill the specific requirements of WSN	- The proposed framework is not evaluated - TDMA protocol needs to be evaluated
[172]	- The proposed architecture is implemented - Apps were build with API in the data collection process	- Detail working of API with heterogeneous data sources are not discussed - Focus only on energy management issues
[173]	- Managing mobility and energy consumption in IoT networks	- No discussion how security and fault management will be encountered
[174]	- Focused on security management	- Details of main processes are not presented
[175]	- The proposed architecture is implemented - Focus on security management solution	- Detailed discussion on implementation is missing - Specific use case consider to implement the framework
[176]	- Discuss the SDN controller for managing industrial process	- No implementation details are provided
[177]	- Focused on security feature to manage IoT networks	- The proposed framework is not implemented

overload conditions such as the heavy number of requests sent to the SDN controller.

RQ4: What scalable solutions can be offered by SDN-based frameworks to manage IoT networks? Author in [158] discusses the scalable middleware solution for interoperability across heterogeneous devices that serve in various application domains such as discovery protocols to manage IoT devices and context-aware IoT applications. Hakiri et al. developed a published subscriber-based scalable middleware strategy in [163]. on the network layer protocols, the authors suggested that 6LowPAN protocols between MediaAccess Control (MAC) and IPv6 work on systems with restricted resources and proposed scalable routing protocol ROLL (Routing for Low Power and Loss Networks) for low-power devices. However, the proposed framework is not implemented to handle IoT management challenges using SDN layers. Authors in [170] discuss the efficient, scalable solutions in terms of the controller to schedule flows rules according to device requirements.Table-9 clearly shows that the majority of the middleware-based SDN framework solutions address IoT network's scalability challenges; however, most of the proposed solutions are not implemented.

RQ5: How can SDN-based frameworks enable efficient power consumption in IoT networks? The computation and security parameters create energy challenges for IoT devices. In [169] a dynamic energy distribution framework is proposed for large-scale IoT in eHealth applications. The proposed framework focuses on energy challenges of IoT

networks concerning dynamic security and forwarding policies to manage the IoT network. The authors in [172] implement an energy-efficient framework that included a cloud data center, an industrial machine with AVG, IWN, RFID scanner, conveyor, and so on. According to the findings, the IIoT SDN-based design uses less energy, is more stable, and allows for autonomous industrial decision-making. Table-9 clearly shows significantly fewer efforts are made to address the energy-efficient middleware-based SDN framework to manage IoT networks.

VII. OPENOW ADAPTATION BASED MANAGEMENT FRAMEWORKS

The control plane (CP), southbound interface (SBI), and data plane (DP) are the essential elements of an SDN architecture [178]. The application plane comprises network applications that specify the rules and instructions that govern the network logic using the exposed northbound APIs. These instructions are translated to the control plane by the northbound API interface, which offers fine-grained control over the forwarding nodes and provides many network services, such as routing, monitoring, load balancers, and firewalls [179]. These applications are either embedded in the control plane (e.g., optimization of routing, management and monitoring of networks, security, traffic engineering, and control of QoS) or located on a proxy server (e.g., firewall and firewall control). The control plane consists of one or more controllers that, through the Southbound APIs interface, forward the instruction sets and policies specified by network applications to the data plane [180]. OpenFlow [181] is the first and most prevalent SDN flow control protocol, which is now the de facto standard for SDN switch control. In order to allow the controller to have direct access and control of the data forwarding network devices, it plays the function of the southbound interface. The Open Networking Framework (ONF) standardizes OpenFlow to cope with the varied life and high latency of applications and decrease management complexity. Flow control systems such as forwarding and control element separation (ForCES) and protocol-oblivious forwardings (POF) are examples of southbound [182]. Similar to the OpenFlow flow tables, ForCES uses logical function blocks (LFB) to provide networking functionality, such as IP routing, to data forwarding devices. In OpenFlow-SDN, the controller has visibility of the global network state over the network. The forwarding rules (flow entries) can be proactively configured on each linked data forwarding unit's flow tables. However, it has been used to implement flow tables due to the high wildcard lookup efficiency of Ternary Content-Addressable Memory (TCAMs) [38]. OpenFlow protocol for SDN is designed for traditional networks. The protocol maintains flow tables across the network and populates the tables with the decision from the central SDN controller. This design is not suitable for constrained IoT networks which usually run over the 6LowPan protocol stack. Therefore, in the existing literature, there are a number of efforts to adapt the OpenFlow operation and table and message structure better to accommodate IoT networks' requirements.

A. SENSOR OPENFLOW

1) Motivation

WSN are application-specific and, due to network topology changes, they are challenging to handle. By adopting the Open FLow protocol, the authors in [183] suggested an SDN-based architecture for IoT to address these challenges.

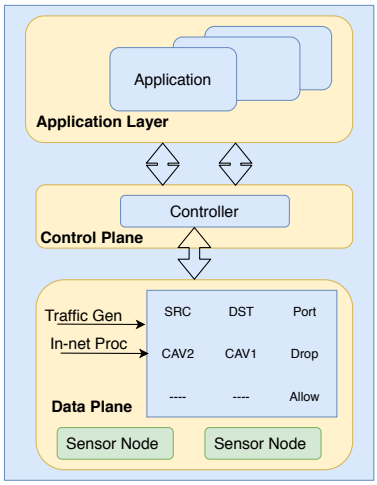

FIGURE 29. Sensor-OpenFlow.

2) Proposed Framework

The SDN-WSN, as shown in Fig. 29, was introduced by Lou et al. [183], with a simple split between the control plane and a data plane using OpenFlow as an agreed protocol for interaction between the two planes. The data plane has nodes that perform the flow table-based packet forwarding. The WSN is very versatile, flexible, and easy to manage by incorporating SDN into WSN. Since OpenFlow has nevertheless been designed as a wired network protocol, it needs some tweaking to make it suitable for wireless networks. This has been the task of Lou et al. [183] with their proposed OpenFlow Sensor system. The Sensor OpenFlow control channel is similar to the OpenFlow control channel. In SDN OpenFlow, the channel is out of band, which is not realistic for WSN, and the Sensor OpenFlow channel is hosted in a band, which means WSN has to carry additional control traffic. This becomes quite an overhead since control traffic in WSN is already significant due to high network dynamics and results in the WSN getting overloaded rather quickly. WSN typically does not process data as it arrives but instead aggregates data and then processes it to conserve network resources and bandwidth. The first solution in this regard is to rewrite flow tables. The second option would be to augment WSN to handle IP traffic so that the control channel can work both with IP and non-IP-based traffic. For IP traffic, the Sensor OpenFlow channel is equipped with a superimposed transport protocol over the WSN. If an operator chooses WSN with IP, then sensor OpenFlow channels are self-supplied.

3) Critical Analysis

The authors have not provided any details on how they are addressing the major challenges of IoT management, such as load balancing, energy management, and so on. In addition to this, implementation and evaluations of the proposed architecture have not been performed. Without formal and detailed studies on the architecture's performance under different scenarios, and working conditions, the proposal is merely a conceptual exercise.

B. FRAMEWORK FOR IOT VIRTUALIZATION VIA OPENOW

1) Motivation

Establishing an IoT ecosystem through networking and resource sharing in configurable and dynamic networks among many physical entities will lead to ambient computing and pervasive intelligence. The vision of achieving technology as a service can be fulfilled through the collaboration between the IoT and OpenFlow.

2) Proposed Framework

The proposed framework by [184], consists of four layers, i.e., connectivity layer, access layer, abstraction layer, and service layer. These layers form interfaces among services and units through network virtualization. This layer also verifies the availability of physical resources and network infrastructure. The access layer consists of the topology specification, activation of the network, and domain formation. It also manages link setup, intra-inter domain communication, scheduling, and packet transmissions between flow sensors and IoT gateways. One of OpenFlow's core features is adding virtual layers to an architecture, leaving the actual infrastructure unchanged. Thus, for various networks, a virtual connection can be generated, and a common platform can be built for different communication systems. The storage and maintenance layer includes data storage and supervision, and the service layer provides information resources and business management and operations.

3) Critical Analysis

The framework performance assessment is conducted for three different scenarios: internet communication, intra-domain communication, and cross-domain communication. In all of these scenarios, a significant increase in performance can be seen.

C. CLOUD-ENABLED SECURE IOT ARCHITECTURE THROUGH SDN

1) Motivation

The expected deployment of IoT technologies in several real-world applications, such as surveillance, transport, and environmental manufacturing, could be seriously undermined by cybersecurity threats to low-cost end-user devices. Also, the enormous quantity of data these devices generate creates new problems with efficient collection and analysis of data, decision-making, and behavior execution. The authors in [185] proposed CENSOR, a new cloud-enabled secure IoT network architecture based on the SDN paradigm, to tackle these issues.

2) Proposed Framework

The proposed architecture of [185] is shown in Fig. 30. The architecture is divided into four layers, i.e., application layer, control plane, and data plane. The data plane is responsible for controlled sensors, and actuators use a Trusted Platform Module (TPM) responsible for the safety and essential services related directive from the IoT controller. The control plane has several centralized SDN controllers that manage various IoT environments in several situations, such as security management, topology management, resource management, IoT service management, traffic, and device management. The modules of multiple controllers also communicate with Cloud data centers for various purposes such as data analytics, NFV based integration of security services, etc. The application plane consists of different IoT services responsible for implementing the business logic and data storage application-level policies.

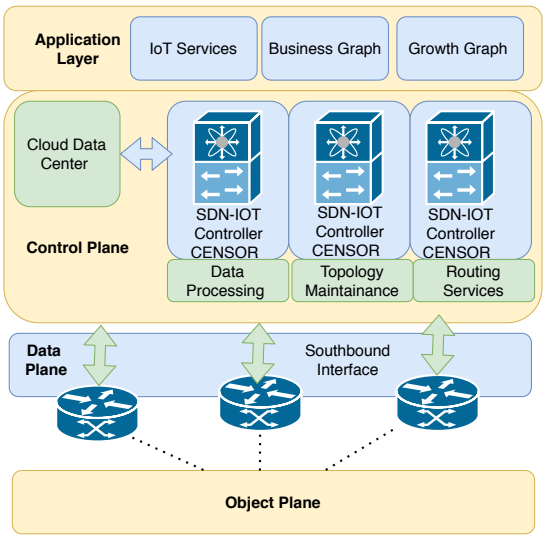

FIGURE 30. CENSOR: Cloud-enabled secure IoT architecture over SDN paradigm.

3) Critical Analysis

The proposed architecture is based on a cloud-enabled secure IoT SDN paradigm. The analysis report of the proposed framework shows that the framework is resistant to various security threats. The authors never discuss the attestation process between the control plane and data plane and the deep packet inspection algorithm.

D. SDN FOR WIRELESS MOBILE NETWORKS

1) Motivation

SDN is widely used in most computer networking application architectures such as data centers, private clouds, public

clouds, etc. However, some of the legacy network architecture is now in the removal stages, such as cellular networks 2G, LTE, etc. These legacy systems can be revived with the adoption of SDN-based solutions. The legacy cellular network consists of Gateway GPRS Support Node (GGSN), Serving GPRS Support Node (SGSN), Base Station Controller (BSC). These elements are responsible for mobility and session management of the mobile stations, and the station controller also provides functions such as encryption, decryption, and authentication.

2) Proposed Framework

Authors in [186] proposed a new architecture for the 2G legacy network architecture with the SDN-based approach that adopted noncellular access technology or domain such as IoT networks. The proposed architecture based on OpenFlow protocol takes the existing standard GPRS as a baseline removing the GGSN and SGSN nodes from the legacy cellular networks with new nodes in the architecture. The new nodes consist of subnodes: ePCU (enhanced Packet Control Unit), SDN controller, vGSN (virtual GPRS Support Node), and OpenFlow-based forwarding core. The sub-node ePCU of the new node is responsible for understanding the GPRS protocol packet and separating the signaling from user plane data. This node is working as a special kind of OpenFlow forwarder. The other sub-node vGSN is responsible for processing the signaling messages, mobile station or BSC, and assisting during authentication procedures or session management procedures. vGSN communicates with the SDN controller, which works as an OpenFlow controller through the Gb interface.

3) Critical Analysis

The authors implement the proposed architecture. The setup was composed of Sysmocom SysmoBTS, a relatively inexpensive 2G (850/900/1800/1900 MHz) BTS. The transport core controlled by the SDN-based OpenFlow controller. The transport core itself is based on OpenFlow compliant forwarder responsible for executing MAC tunneling according to OpenFlow rules set by the controller. However, the access edge forwarders (ePCU) examine the IP header and the access-specific header (e.g., GPRS-specific protocols). The external network edge (e.g., Internet uplink) also examines the IP header to select the correct tunnel for a particular mobile station. The new architecture is just removing SGSN, GGSN nodes from the legacy cellular network. In the new architecture, the SDN OpenFlow controller is in charge of transport core and connectivity orchestration. The proposed architecture is hypothetical because the authors never discuss implementation details and never discuss the OpenFlow protocol details. No justification is provided that the proposed solution play any vital role in any real-time problem

E. CENTRALIZED ARCHITECTURE FOR COMMUNICATION NETWORK BASED ON SDN

1) Motivation

The rapid growth of the IoT has encouraged the vigorous development of new services in distributed networks while at the same time suggesting higher differentiated performance standards for the communication system. Power internet of things has introduced challenges of performance requirements due to power ecosystem complexity, limited capacities of connected devices in power systems, threats, attacks, etc. There is a need for SDN-based management in the power internet of things to overcome these challenges.

2) Proposed Framework

Authors in [187] proposed architecture that consists of three layers. The top layer is the controller cluster, composed of three cluster management layers responsible for maintaining the overall network stability and completing the functional task with the Root controller and information synchronization. The local root layer manages complex local services. In the last layer, the local controller communicates with the switch using the OpenFlow protocol. The next layer is the FLowvisor network virtualization platform, responsible for providing proxy between the lower and SDN cluster controller FlowVisor generates rich "slices" of network resources. To reduce the single point failure in the controller, centralized cluster management is implemented in which the root controller takes information from the local controller.

3) Critical Analysis

The proposed architecture is very abstract and needs a detailed explanation of the algorithms involved. The authors have not discussed the implementation and no formal evaluations have been conducted to test the different scenarios.

F. OPENFLOW ENABLED POLICY-BASED IOT NETWORK SECURITY

1) Motivation

The implementation of the SDN paradigm in networking improves the traditional architecture of computer networks. The adoption of the SDN paradigm in IoT has increased rapidly in the recent past, but this adaptation often presents challenges in the IoT domain due to high volume and network traffic rates, variations in the characteristics of IoT systems and computer networks, and limited resources in the underlying network framework.

2) Proposed Framework

In [188], authors proposed an IoT-NETSEC framework based on SDN technology consisting of the following building blocks: device policy repository, IoT device registration, security flow role installer, statistic collector. The proposed framework monitors the traffic of IoT devices across the network to ensure three basic rules: only approved communications are allowed and everything else is denied, monitor the network traffic, and protect the IoT device against three attacks such as port scanning DOS, DDOS. The proposed

framework can be used as a security as a service application in an IoT domain. The device policy repository is responsible for containing dynamic policy documents. This means that the policy document in this module is changeable according to the situation or reconfigure. The parameter to making policy is the device name, type, and set of flow rules for IoT nodes communicate with the SDN controller IoT device registry responsible for registering the IoT device with a proper mechanism to communicate with the SDN controller through IP address Port number. The device policy is of the particular node is implemented with the help of the device policy repository. Security Flow-rule Installer is responsible for providing routing and non-routing flow entries. This module parses the security ruleset from the IoT nodes and then creates related security flow rules for traffic monitoring purposes and installs the flow entries in the SDN Switch. Before deploying the security flow entries, ensure whether they are already relevant flow entries rule at the switch in the network. Statistic collectors collect the packet from the IoT nodes associated with monitoring flow entries in the SDN switch in the IoT network to fine-grain the monitoring ability through statistical knowledge. Statistics analyzer is responsible for getting information from the statistic collector and analyzing the packet in deep about the traffic flow both statistic collectors connect.

3) Critical Analysis

The proposed architecture's implementation is evaluated with an OpenFlow-enabled switch. It runs on a dual-core 2.4 GHz Intel Xeon processor connected to the controller through a 1Gbps shared link with a ping delay of 0.5 ms. The proposed solution implementation testbed is very limited; the dataset for experimentation is not trustworthy; the authors never discussed the details for choosing the test data and training data for analyzing. The proposed framework focuses only on security features with a basic Machine Learning (ML) approach that may show a better result with other methods applied like deep learning.

G. SDN-BASED SECURITY FRAMEWORK IN DISTRIBUTED GRID

1) Motivation

SDN is emerging as a new model for the next decade's network infrastructure. The separation of the control plane and the data plane inside the SDN brings the versatility to use complicated software programs to handle, configure, protect and maximize network resources. Security point of view SDN can collect information from network devices and allow applications to program forwarding devices, unleashing a powerful proactive and smart security policy technology. Unlike conventional protection solutions based on a static firewall programmed by an administrator, such as the Intrusion Detection and Prevention System (IDS/IPS), these functions enable the incorporation of security tools that can be used in distributed scenarios. This network's programmability can be implemented to create a modern networking channel for the IoT.

2) Proposed Framework

The proposed architecture shown in Fig. 31 by [189] is based on an Opendaylight MD-SAL Akka-based clustering solution. The authors proposed a routing algorithm with distributed cluster SDN routing protocol, which can be used to facilitate SDN-based inter-domain collaboration, to select a suitable route between nodes connected to the cluster. The proposed algorithm automated the domain clusters. The OpenFlow protocol specifies control messages for creating this application, allowing the SDN controller to create a stable link to network devices, read their current status and install forwarding instructions.

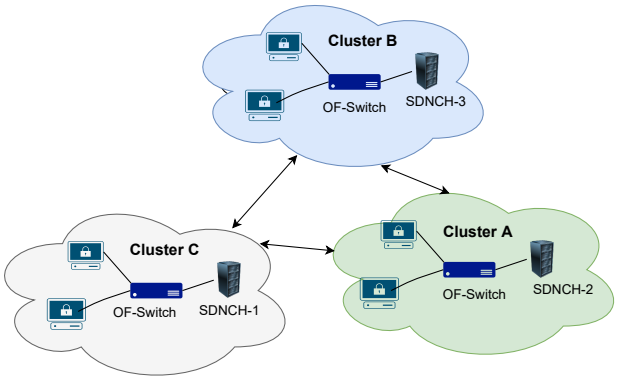

FIGURE 31. SDN-Based Security Framework in Distributed Grid.

3) Critical Analysis

The proposed framework is tested on a very small testbed. Moreover, the framework working details are missing in the proposed algorithm. Authors only focus on the security parameter related to the routing algorithm.

H. DISCUSSION OF RESULTS

At the moment, OpenFlow is the most extensively utilized SDN approach. In an SDN architecture, OpenFlow is a method that standardizes how a controller communicates with network devices. The OpenFlow protocol maintains flow tables in the network and populates the tables with the SDN controllers. This design is not sufficient for IoT networks that usually operate over the 6LowPan protocol. Therefore, we choose those papers in which research efforts are directed to adopt the OpenFlow operation for better accommodating IoT networks' requirements with a proposed framework or implementation of the proposed solution. We also selected some papers that can be used as a building block to understanding how OpenFlow adaptation addresses IoT management issues. Addressing the management issues of IoT OpenFlow taxonomy, according to the literature assessment, is also essential. We summarize the critical rationale

TABLE 11. Summary of OpenFLow Adaptation based Solution addresses IoT Management Challenges

Exiting Work	Fault tolerance	Energy	Load balancing	Security	Scalability	Implementation
[38]	✓		✓	✓		
[178]	✓		✓			
[179]	✓		✓		✓	
[180]	✓		✓		✓	
[181]	✓				✓	
[182]			✓	✓	✓	
[184]			✓	✓		✓
[185]	✓		✓			✓
[186]		✓			✓	✓
[187]	✓		✓		✓	
[188]		✓		✓	✓	✓

TABLE 12. Summary of OpenFlow based IoT solution

Existing work	Merits	Demerits
[183]	- OpenFlow adaptation is proposed to manage IoT networks	- No evaluation of OpenFlow adaptation proposed
[184]	- The Framework performance assessment is conducted for three different scenarios - Result show significant performance increase	- Proposed framework is based on too many layers that cause increasing processing delay
[185]	- The proposed architecture is implemented - Only security is discussed	- Not discussion on the details of attestation process between control plane and data plane
[186]	- The proposed architecture is implemented - Considered only cellular networks	- Discussion of implementation is missing
[187]	- Discuss some algorithms for single point failure countermeasure	- The proposed architecture is not implemented - Proposed architecture is very abstract and lack implementation details
[188]	- The proposed architecture is implemented	- The implementation test bed is very limited - Focus only on security features
[189]	- The proposed secure routing algorithms are implemented	- The implementation test bed is very limited - Focus only on security features

supporting the OpenFlow taxonomy that addresses the IoT challenge by answering the following research questions.

RQ1: How can SDN-based frameworks provide efficient security solutions to manage IoT network-related security issues?

Authors in [180], [181] highlight the importance of security access policies at the controller level. OpenFlow can address the security challenges of IoT networks in terms of network degradations, network throughputs, etc. The control plane consists of one or more controllers that forward the instruction sets and policies specified by network applications to the data plane through the Southbound APIs interface. Table-12 suggested minimal effort has been made to address the management challenges of IoT, and many of the proposed solutions lack detailed features discussion. Authors in [188] proposed monitor modules in the proposed framework to monitor the traffic of IoT devices across the network to ensure three basic rules: only approved communications are allowed and everything else is denied, monitor the network traffic, and protect the IoT device against three attacks such as port scanning DOS, DDOS attacks

RQ2: How can SDN-based frameworks provide effective fault tolerance management solutions to large-scale IoT networks? To address the fault tolerance difficulty of managing IoT networks, the authors in [178], [179] describe the control plane and data plane roles. The northbound API interface, which allows fine-grained control over forwarding nodes and numerous network services such as routing, monitoring, load balancers, and firewalls, governs the control plane. These applications are either built into the control plane (for example, routing optimization, network administration and monitoring, security, traffic engineering, and QoS control) or hosted on a proxy server. Table12 suggested that to maintain the fault tolerance, other IoT management parameters such as scalability and load-balancing must be taken into consideration.

RQ3: What are the potential solutions regarding load balancing in SDN-based frameworks to manage IoT networks?

Ambient computing and pervasive intelligence will be enabled by establishing an IoT ecosystem by networking, and resource sharing among many physical elements is configurable and dynamic networks. Through collaboration between IoT and OpenFlow, the concept of technology as a service can be realized. Authors in [184] proposed a framework that focuses on load balancing technique in terms of data storage, and the load-balancing algorithm is applied in a testbed of multiple virtual storages. Existing literature suggests that many implemented solutions and research efforts have already been made in this regard to manage the IoT environment.

RQ4: What scalable solutions can be offered by SDN-based frameworks to manage IoT networks?

Scalable solution to address the IoT management challenges with OpenFlow taxonomy is generally correlated to security, fault tolerance, load-balancing parameters [178],

[179], [184]. Literature suggests the most of the proposed frameworks are scalable. The majority of the proposed solution working on controller ends have algorithms for security, load balancing, and fault tolerance.

RQ5: How can SDN-based frameworks enable efficient power consumption in IoT networks? SDN is widely used in data centers, private clouds, public clouds, and other fields of computer networking. However, certain older network design is being phased out, such as 2G, LTE, and other cellular networks. With the implementation of an SDN-based solution, these legacy systems can be brought back to life. Authors in [186] highlights the key energy parameters involve to address the management challenges of IoT in terms of security algorithms. The rapid growth of the IoT has prompted the rapid development of new services in distributed networks while also implying higher specialized communication system performance demands. Because of the power requirements of the IoT, there have been specific performance difficulties. To counter this, authors in [188] proposed a security policy repository in order to manage the energy challenges of IoT networks.

VIII. BLOCKCHAIN-BASED SDN MANAGEMENT FRAMEWORK

Blockchain is a distributed ledger of continuously growing data in chain order in which each block is secured using a cryptographic algorithm [190]. It enhances management by verifying data such as digital content management [191] [192]. Blockchain can be used to store data, verification authentication, currency transaction, etc. The concept of blockchain is introduced from the Bitcoin crypto-currency system launched in 2008 by Satoshi Naka-moto. Blockchain may typically be used to provide security services. For example, applications have already emerged from blockchain-based identity providers, voting systems, financial services and supply chain management, etc. Blockchain seems to be the driving technology contributing to a significant part in IoT technology's [193]–[196].

Blockchain is essentially a perfect complement to IoT with improved interoperability, privacy, security, reliability, and scalability [197]–[199]. We have focused on existing literature that directs their focus towards integrating blockchain with the SDIoT framework.

A. A BLOCKCHAIN-BASED TRUST FRAMEWORK FOR IOTS

1) Motivation

IoTs are anticipated to open up challenges for researchers and industry vendors for better coordination between different IoT networks. Cross-platform collaborations are required for sharing data with other IoT applications.

2) Proposed Framework

The authors in [200] introduced a decentralized trust system named IoT passport for cross-platform collaboration between IoT applications based on blockchain technology. This passport is essentially used for authorization, authentication, and trust. Through smart contracts, data security and privacy apply to secure data communication between applications and nodes. The database for the IoT passport is responsible in the form of an identity registry for each IoT node. The IoT repository consists of an intelligent contract known as the IoT Passport contracts, consisting of identity mapping, identity registration, and revocation. The user-defined policies module is responsible for ensuring policies that trigger a given condition and are responsible for further action during node interaction. The access control policy module is responsible for providing an authorization mechanism for cross-platform communication nodes. This access control policy written in a smart contract called the trust rule contract with identity authentication, access control, and trust between nodes. The incentive policies module makes policies for miners who are involved in the transaction operation. The agreement is also written in the form of the smart contract, which finally gives some reword based on miners' efforts.

3) Critical Analysis

The proposed structure is based on five fundamental principles associated with the intelligent contract. The authors provided no analysis of the proposed architecture. Moreover, the authors have not provided a profound discussion of the core blockchain theory, and no specific context-aware design control algorithm is submitted.

B. BLOCKCHAIN-BASED FRAMEWORK FOR EDGE AND FOG COMPUTING

1) Motivation

Recently, efforts have been made to integrate Edge, Fog, and cloud-based services to support IoT applications, but they come with unique security, resource management, and multi-application execution limitations. To address these limitations, a framework called FogBus that supports end-to-end IoT-Fog-Cloud integration to ensure data integrity, data confidentiality, reliability through blockchain is proposed in [201].

2) Proposed Framework

Authors in [201] proposed architecture that comprises IoT devices, Fog gateway nodes, Fog infrastructure, cloud infrastructure, broker nodes, general nodes, and repository nodes. Fog gateway nodes are the entry point for IoT devices to communicate with Fog computational nodes via FogBus terminology. FogBus is responsible for supplying IoT devices authentication credentials service, conveying service expectations, obtaining service results, handling IoT device requests effectively. Fog gateway nodes are responsible for fast and dynamic communication with accessible Fog nodes through COAP or SNMP protocol. Fog Bus simultaneously communicates several heterogeneous Fog computer nodes that communicate nodes with broker nodes and general repository nodes. Fog computational nodes start data processing and

find the best available tools for the available repository nodes in the local area of the requested IoT system. General computing nodes essentially supported various network functions in virtualization such as firewall services, network service managers to control service quality, etc. Repository nodes are responsible for facilitating data exchange, replication, recovery, and secure storage. Repository nodes provide interfaces for instant access and historical data analysis. They maintain the meta-data of different applications, including application models, runtime specifications, and dependencies. Fog infrastructure is overwhelmed by traffic or does not provide the required service. Fog infrastructure interacts with the cloud data center to conduct the necessary service through cloud services through cloud data centers. In combination with Fog repository nodes, it enables comprehensive data storage and distribution such that data access and processing become location-independent.

3) Critical Analysis

The authors introduce the proposed architecture called the Sleep Apnea prototype. The embedded blockchain function in FogBus architecture is very generic. The security feature implemented with blockchain aid increases computational time in resource management, security mitigation steps run time framework migration.

C. SDN AND BLOCKCHAIN-BASED TRUST MANAGEMENT FOR IOT DEVICES

1) Motivation

In the IoT domain, it is challenging to recognize devices that are vulnerable to the environment due to a lack of required knowledge and available solutions. An SDN and blockchain-based trust system using an SDN controller to establish a trust level through a trust score based on the record in a blockchain known as StewARD is proposed in [202].

2) Proposed Framework

Authors in [202] proposed an architecture that consists of a Blockchain layer, analyzer, frontal, and controller. The blockchain layer is responsible for tagging the devices as good behavior, bad behavior, malicious behavior, bind, and leave. Good behavior is tagged when the chain of that device in blockchain reports that the device is behaving according to the rule. Bad behavior is when the device chain history shows some deviation is detected. Malicious behavior is tagged when the chain history shows some abnormal activities of traffic according to defined rules. Bind is responsible for joining a new device to the controller before connecting to a slice. Such pairing aims to prevent fake or malicious home controllers from reporting on devices that they do not manage. Leave terminate the device's connection to the controller permanently, the same as the concept of proof of burn in the blockchain. The analyzer is responsible for continuously analyzing the traffic in raw data from the blockchain to trust the IoT device. With the help of assessing the report, the modules find whether the new or unknown device can harm the network. The front layer is responsible for interaction between the controller and analyzer. Through this middle layer, the controller can access the trust score of each device to decide whether the device connects to the network slice or not. The information on the trust score can be pulled by the controller from the frontal, middle layer.

3) Critical Analysis

The proposed architecture is based on blockchain to compute a trust score and provide this report to the controller. On behalf of this report, the controller dynamically (dis) connects a device (from) to a slice labeled with a certain trust level. The proposed architecture may have the potential to deal with the surface attack. The authors missed the discussion on the details of the algorithms used in the proposed architecture.

D. FORENSICS ARCHITECTURE IN SDN-IOT USING BLOCKCHAIN

1) Motivation

The IoT domain faces challenges in digital forensics, including data integrity, deletion of proofs, or modification to resolve those challenges. Blockchain technologies, even if used, can present weak attack detection and sluggish processing. SDN-IoT provides an efficient forensic architecture to overcome these challenges that create a Custody Chain (CoC) with blockchain technology.

2) Proposed Framework

Authors in [203] suggested SDN-based IoT architecture, where controllers are implementing flow-table switch rules for three different traffics, Voice over Internet Protocol (VoIP), File Transfer Protocol (FTP), or HyperText Transfer (HTTP) Protocol. The architecture uses a Linear Homorphic Signature (LHS). The parsing of the message includes Flow-Mod, Packet-In, Stats-Reply, and other necessary packet features. The controller feature analyzer module is responsible for the feature extraction of the entry packet based on the attribute's value. The authentication module in the controller authenticates the device using the LHS algorithm, which considers the authentication of a single IoT device with an Elliptical Point. This module contains flow rules based on the type of traffic, protocol, or port number. Only three types of traffic with the help of flow table rules are permitted or disclaimed in the proposed architecture. There are a variety of flow entries for each change in the proposed system. Before treatment, each switch verifies all three traffics and the corresponding port numbers. The change discards invalid traffic with the wrong port number. These three traffics checked with port numbers as unauthorized users access the network using an invalid port number. The authors proposed two algorithms. One algorithm focuses on the process followed for switches implemented in the control plane. The other algorithm targets the process followed for the controller in the SDN controller using the Neuro Multi-Fuzzy model in the controller for classifying the legitimate user involved in the

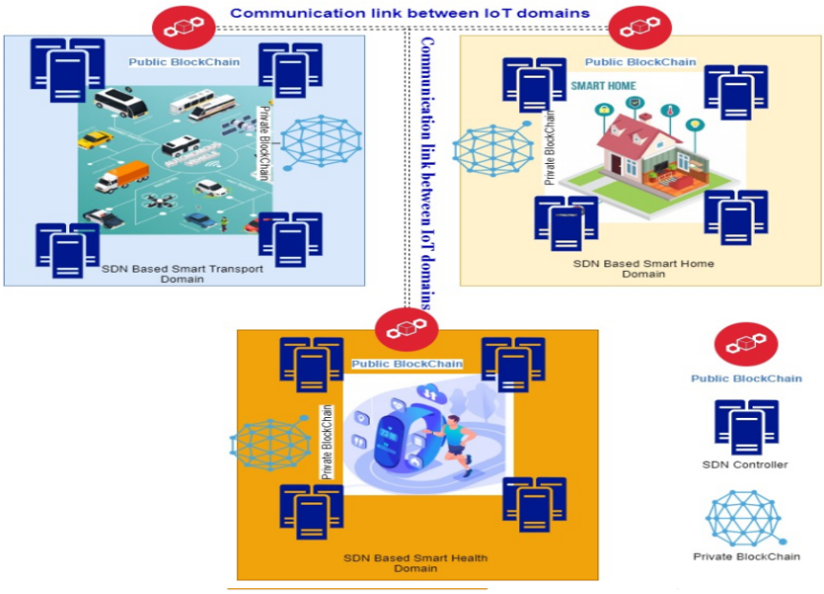

FIGURE 32. An Energy-efficient SDN Controller based on Blockchain.

network. The devices are authenticated from the blockchain, and are analyzed on the neuro multi-fuzzy model.

3) Critical Analysis

The Network Simulator Version 3 (NS3) forensic architecture in SDN IoT is developed. In NS3, the blockchain concept was integrated into IoT based on SDN. In order to implement the blockchain concept in SDN, this architecture was created using a Bitcoin coding framework in NS3. The blocks in the blockchain are generated in 10 sec on average, which can be improved for different scenarios.

E. BLOCKCHAIN ENABLED ENERGY-EFFICIENT SDN CONTROLLER ARCHITECTURE FOR IOT NETWORKS

1) Motivation

There are long-standing challenges in the IoT market, such as security, comparability, energy consumption, and device heterogeneity. Security and energy factors play essential roles in data transmission across IoT and edge networks. The merger of blockchain and SDNing (SDN) can resolve the energy and security parameter issues in IoT networks.

2) Proposed Framework

Authors in [204] proposed architecture, as shown in Fig. 32, distributed network management for IoT devices implemented using an IoT-tailored blockchain and SDN controller in a cluster structure. The architecture in which the SDN controllers linked to a single blockchain can communicate IoT devices. The proposed architecture's key objectives are to enhance the security of IoT communication and reduce energy consumption. They presented an algorithm for energy efficiency and security. Private and public blockchains are used in the proposed architecture, optimized for the IoT network. The proposed algorithm is based on the configuration of the cluster and the limitations of IoT devices in terms of energy and computation. The algorithm utilizes the blockchain security features to improve security in line with the energy efficiency criteria and an SDN controller for the process of authentication and verification in each cluster.

3) Critical Analysis

The proposed architecture shows a significant impact on reducing energy consumption and increasing communication protection between IoT devices. The architecture missed addressing the load balancing and resource management issues in IoT networks, which could have been accommodated for an effective solution.

F. SDN-BASED DISTRIBUTED BLOCKCHAIN ARCHITECTURE FOR IOT

1) Motivation

The recent growth of the IoT and the subsequent proliferation of data volumes created by intelligent devices have contributed to outsourcing data to specified data centers. However, consolidated data centers, such as cloud computing, can not continue to handle these massive data stores desirably. In conventional networking architecture, there are several problems due to the exponential increase in diversity and the number of devices not built to link to the Internet. Provide high availability, data distribution in real-time, scalability, protection durability, and low latency. A blockchain distributed cloud system with an SDN controller can solve these problems.

2) Proposed Framework

The proposed architecture in [205] is based on three phases. This model monitors and parses important OpenFlow messages from OpenFlow packets to create an overall network view in the first phase. In the second phase, the data set was analyzed, and the state of routing topology extracted, and Metadata features sets for building a traffic flow topology grid. The proposed architecture maintains the Metadata's topological status, flow design rules for outbound flows, store transmission of inbound packet headers, etc. A particular metadata flow validate the permissible metadata values collected over the flow and management strategies duration in the third stage. The model flags knew attacks by the manager strategies, despite being the most specific flow activities conducted over time to detect potentially malicious activity. When the model finds new flow behavior, it does not trigger an alarm: it triggers alarms when it recognizes unreliable entities that change an existing flow or flow behavior, which challenges a specific security policy.

3) Critical Analysis

The author provides the network with a programmable controller to ensure scalability and durability with high availability. The proposed architecture uses cloud and fog nodes for data collection, and blockchain is used to protect data transfer transparency. The data processed at the server's end is secured, enhancing the possibility of confidential data leakage. Fog nodes for protecting data transfer from cloud to IoT nodes. The blockchain functionality is used for cost-effective access management systems. The proposed architecture will greatly minimize end-to-end delays for IoT applications, machine resources, and core network traffic loads relative to conventional IoT architecture.

G. BLOCKCHAIN-BASE SDN MODEL FOR IOTS

1) Motivation

The combination of SDN, NFV, and blockchain are capable of addressing reliable communication in IoT environments such as protection, privacy, flexibility, performance, and IoT environment availability. A safe communication platform or channel has been highlighted as a key requirement for efficient communication in IoT systems.

2) Proposed Framework

In [206], the authors proposed a smart condominium framework based on SDN technology and blockchain technology, to improve the protection of IoT environments. The proposed architecture is based on a layered approach: IoT device Layer, SDN controller layer, NFV layer, Middle Layer, or cloud orchestration layer. The IoT device Layer works as a perception layer of the IoT environment, such as sensors, responsible for extracting the information passing this information to the SDN controller. The SDN controller is responsible for routing the particular data to the destination. NFV layer provides different network functions such as routing, security, etc., to the framework in the form of a distributed package. The cloud orchestration layer is responsible for putting the data on the public blockchain. The proposed architecture is implemented with the topology of 50 network nodes with nine access points (APs).

3) Critical Analysis

A distributed, secure SDN-IoT model based on blockchain as shown in Fig. 33 was propsed by [206]. The study proposed a CHS (Cluster Head Selection) algorithm that selects CH(Cluster Head) with the highest energy optimally. The SDN controller continuously monitors and manages IoT device information across the entire IoT network; it also detects possible attacks on the network system; it enhances scalability and flexibility issues. NFV then supplies a virtual platform to the SDN-IoT-enabled physical environment and saves money, extending the entire network's lifetime. Distributed blockchain also provides ample security and privacy; it efficiently identifies and mitigates cyber attacks in the proposed scheme.

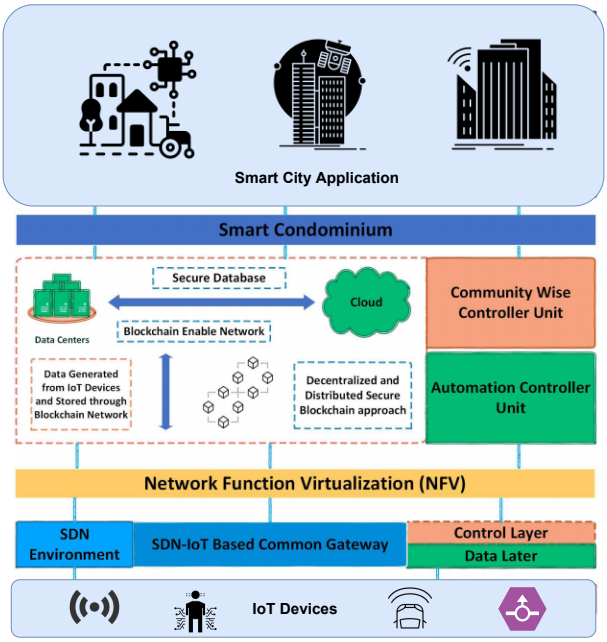

FIGURE 33. Blockchain and NFV for Smart Condominium.

H. BLOCKCHAIN-BASED SECURITY FRAMEWORK FOR SDNS

1) Motivation

Existing literature suggests that in SDNs, the main danger is the single point of failure. Any failure of the controller would affect the network's overall functionality, as the primary objective of the attackers was to compromise the controller. In [207], the authors proposed a security model to ensure compliance with enhanced security based on blockchain technology between instances of the SDN controller.

2) Proposed Framework

The authors proposed a control plane security algorithm and choose to deploy security models using the OpenDaylight SDN controller. The proposed framework uses the open-source blockchain project hyperledger fabric with an adaptive consensus module to build the underlying protection mechanism. Their modular architecture allows the controller and blockchain providers' core services to be combined while maintaining the performance scale.

3) Critical Analysis

The architecture proposed provides a general overview and lacks information on how the various components function and communicate with each other. There is no comprehensive algorithmic work presented because the architecture is laid out in general layers. The authors provided no implementation and evaluation of the architecture as well.

I. DDOS BOTNET PREVENTION USING BLOCKCHAIN

1) Motivation

The study [208] addresses the increasingly growing number of IoT devices, which at the same time leads to networking, protection, management problems, and the possibility of being part of a botnet to launch a DDoS attack. According to the researchers, the Internet of Everything (IoE) leads to more and new problems rather than solving existing ones. Therefore, they proposed new techniques to protect IoT networks against DDoS attacks.

2) Proposed Framework

The proposed framework integrates a blockchain SDN controller to manage the distributed nature of IoT devices efficiently. The proposed framework consists of three modules: Security Policy Module (SecPoliMod), Controller Module (ConMod), and LogModule (LogMod), in which SecPoliMod and ConMod are primarily programmed to prevent the use of IoT devices as botnets, while LogMod controls network traffic for the devices in order to ensure their legitimacy. To implement security policy and differentiate between legitimate and illegitimate connected devices, SecPoliMod relies on the colored coins concept introduced by blockchain technology. If a device is colored, this indicates that the device has met the minimum network link security criteria. However, network traffic flowing from that system will be separated and dropped by the switches before integrating with other network traffic if no label is identified on the device.

3) Critical Analysis

The proposed architecture lacks the discussion on the algorithmic approach and does not provide implementation details that can demonstrate the effectiveness of the proposed framework.

J. DISCUSSION OF RESULTS

It has been discovered that IoT devices generate a large amount of data, which must then be stored and evaluated for analysis to extract new insights. Blockchain has played a significant role in decentralized IoT networks. Distributed Ledger Technology (DLT) and decentralized cryptocurrencies (such as Bitcoin [211], Ethereum [212] etc., and the technology beyond them has become a trending research area in recent years. For every IoT operation (such as create, update, delete, and read) in the blockchain blocks, each data item can be saved as a transaction. Smart contracts can be used to register the identity information of IoT devices in a block with current status and instance information of production, as well as control policies for IoT devices. Table-13 summarizes the efforts of SDN-based blockchain taxonomy to address the management issue of IoT networks.

RQ1: How can SDN-based frameworks provide efficient security solutions to manage IoT network-related security issues? Due to a lack of experience and evaluation in the exiting IoT solution, it is challenging to identify devices that are vulnerable to the environment in the IoT domain. The authors in [202] suggested StewARD, an SDN and blockchain-based trust system that uses an SDN controller to determine a trust level based on a trust score based on a blockchain record. In [203] highlights the IoT domain confronts issues in digital forensics, such as data integrity, proof deletion, or alteration to resolve those problems. Although blockchain is being used as a solution in the existing literature, the technology itself has poor attack detection and processing speed. SDN-IoT presents a forensic architecture that efficiently overcomes the obstacles of creating a Custody Chain (CoC) using blockchain technology. Authors in [213] discuss the blockchain-based security solutions in terms of privacy leakage and selfish mining. Table-13 shows that security features are one of the core themes of blockchain-based SDN solutions to address the management challenges of IoT networks.

RQ2: How can SDN-based frameworks provide effective fault tolerance management solutions to large-scale IoT networks? Few of the existing studies [191] and [192] examine security and resource management in terms of fault tolerance technique using a private and public blockchain method. In [200], the authors explored a use case study in terms of contact aware Access Control for IoT, in which fault-tolerant routing is one of the critical elements, as well as how fault-tolerance mechanisms relate to other key challenges of IoT. The authors in [201] discusses the fault tolerance in three aspects that is Fog computational nodes, Computing Services, and Network topology aspect. Fog computational Nodes maintain a fog level table and a rule-based mechanism to maintain the fault tolerance.

RQ3: What are the potential solutions regarding load balancing in SDN-based frameworks to manage IoT networks? Edge, fog, and cloud infrastructure all work independently in an IoT ecosystem. However, efforts have recently been made to integrate all of these to serve IoT

TABLE 13. Summary of Blockchain based Solution addresses IoT Management Challenges

Exiting Work	Fault tolerance	Energy	Load balancing	Security	Scalability	Implementation
[192]	✓	✓			✓	
[197]	✓	✓		✓	✓	
[199]	✓	✓	✓	✓	✓	
[200]	✓		✓	✓	✓	
[201]	✓	✓	✓	✓	✓	✓
[203]	✓	✓		✓	✓	✓
[205]					✓	✓
[206]	✓		✓			✓
[209]		✓	✓	✓	✓	✓
[210]	✓		✓	✓	✓	

TABLE 14. Summary of Blockchain based SDN IoT Solution

Existing work	Merits	Demerits
[200]	- Proposed generic framework to support heterogeneous IoT application - IoT nodes policies repository is composed of smart contract - Cross platform collaboration with different IoT application	- No implementation of the framework - Overhead due to blockchain deployment
[201]	- Proposed end-to-end IoT Fog and cloud integration	- Proposed framework is implemented in a limited testbed - Missing traffic learning behaviour
[202]	- Proposed framework shows potential to deal surface attack - Crowd sourced reporting by monitoring the device	- Framework missing the details of proposed algorithm
[203]	- Security provisioning based framework	- Overhead noticed using blockchain
[204]	- Shows a significant impact on reducing energy consumption and increasing communication protection between IoT devices. - Presented an algorithm for energy efficiency and security for Private and Public blockchains	- Detail of experiment focus only on security and energy protocols. Other management issues are not address
[205]	- Ensure scalability and durability with high availability.	- Detail of experiment not provided
[206]	- Proposed Framework is implemented. - Proposed Framework utilized cloud orchestration and NFV	- Detail of experiment focus only on security
[207]	- The proposed framework is not implemented . - Presented an algorithm for energy efficiency and security for Private and Public blockchains	- The proposed framework is not implemented - Proposed framework lacks information on how the various components function communicate with each other
[208]	- The proposed framework discuss three security modules (SecPoliMod), Controller Module (ConMod), and LogModule (LogMod) .	- Not provided any information of implementation - The proposed framework is missing details on how the various components interact and function.

applications, but security, resource management, and multi-application execution load balancing remain the key challenges to overcome. To address these concerns, the authors in [201] introduced FogBus. This platform enables end-to-end IoT-Fog-Cloud integration using blockchain to ensure data integrity, secrecy, and reliability in terms of load balancing. Existing literature suggests that fault load-balancing is the critical feature to address the IoT management challenges due to the consensus algorithm in the blockchain-based SDN framework.

RQ4: What scalable solutions can be offered by SDN-based frameworks to manage IoT networks? With exponential development in network management and configuration complexity, SDN has emerged as a promising network model. SDN aims to improve network function efficiency by making network design and operations more dynamic and efficient. Authors in [209] discuss the way to the scalable approach of SDN solution. In [210], the authors discuss scalability challenges when dynamic solutions are required to manage IoT networks. One of the challenges is dynamic networks policies. Table-13 suggests that the majority of the proposed solutions are scalable in nature, but framework discussion shows that the proposed solutions are implemented in very limited testbeds.

RQ5: How can SDN-based frameworks enable efficient power consumption in IoT networks? Energy usage and device heterogeneity are all long-standing issues in the IoT business. In data transmission through IoT and edge networks, security and energy considerations are critical. The combination of blockchain and SDN has the potential to tackle energy and security challenges in IoT networks. The authors in [204] present an architecture based on distributed network management for IoT devices, which is implemented in a cluster structure utilizing an IoT-tailored blockchain and SDN controller. The architecture for communicating IoT devices using SDN controllers linked to a single blockchain. The suggested architecture's primary goals are to improve IoT communication security while also lowering energy

usage. To maximize the IoT network, private and public blockchains are deployed.

IX. LESSONS LEARNED

This section provides the lessons learned from the proposed taxonomy to address the IoT management issues with SDN integration. Moreover, we present the lessons learned from SDN to address the defined IoT management challenges.

A. LESSON LEARNED FROM SDN TO ADDRESS IOT FRAMEWORK'S MANAGEMENT CHALLENGES

It is known that infrastructures built around SDN-enabled IoT units have a tremendous potential [214]. SDN can provide orchestration for network management in the IoT environment by decoupling the control plane and the data plane, including flexibility and programmability in the IoT network. This is the main reason why the SDN-based IoT networks has the potential to address IoT management issues such as fault tolerance, load balancing, etc. Separation of the control and data planes is a vital aspect of the SDN paradigm. It has obvious benefits in terms of network programmability. The control plane can be centralized or decentralized, that helps in developing and implementing dynamic policies at the perception layer, control layer, etc., in IoT networks to address management challenges.

1) Fault Tolerance

Fault tolerance techniques for IoT networks are classified as fault prevention, fault detection, fault isolation, and fault recovery [170]. SDN controllers can enable the design and creation of efficient fault detection techniques for IoT networks due to its centralized view. The IoT nodes send data to the central controller, which can easily detect faults at particular nodes. Once a fault has been identified, the central control can quickly reconfigure the network to circumvent the faulty nodes or routes. Extensive research is needed to design novel fault detection and mitigation algorithms for SDN-based IoT networks.Literature suggests that, most proposed SDN-based management frameworks for efficient fault tolerance management solutions for IoT networks are at the network layer, with solutions for routing protocols, fault detection, reconfiguration, link status, and congestion control mechanisms. Application and service layer protocols receive less attention.

2) Energy Management

Energy is a precious resource in IoT networks, because the deployed sensor devices do not have access to the uninterrupted power supply. In the SDN paradigm, the SDN controller can help schedule network flows, resulting in energy savings. Furthermore, centralizing the network's architecture allows for an aggregation of energy-efficient knowledge. This is one of the most significant issues that will gain significance with the increasing number of IoT devices deployed worldwide. Literature suggests that most proposed SDN-based management frameworks for efficient energy management solutions for IoT networks concerning lightweight cryptographic algorithms, efficient routing mechanisms, efficient scheduling algorithms etc

3) Security Management

Providing and ensuring security services over a resource-constrained IoT network is challenging as traditional security protocols and mechanisms are not applicable in the IoT security domain. Literature suggests that most of the proposed works in the area of SDIoT security frameworks is related to access-list, authentication, authorization, and key management. However, all of the proposed solutions also have another critical issues, i.e., they depreciate the performance energy consumption. Most proposed security solutions are tested on particular use cases. However, attack mitigation or the prevention module are missing in the majority of the proposed frameworks.

4) Load Balancing

Load balancing is considerably eased by the deployment of SDN in IoT networks. SDN creates a centralized view of the network traffic as the data is being transmitted to the controller. This centralized control can thus be used to optimize the traffic load passing through the IoT network. Furthermore, load estimation techniques and algorithms at the controller can assess the IoT network load, influencing the flow traffic in the IoT network. A number of efforts are made to tackle load management problems in IoT in the application layer and the network layer with the help of efficient path selection mechanisms and efficient load shift algorithms.

5) Scalability

The implementation of SDN in IoT significantly simplifies the scalability of IoT networks. To enhance the scalability of SDN-based IoT networks, several studies have been conducted in the past. The control plane was first restructured by scattering controllers horizontally or hierarchically while maintaining unified control over each distributed controller. According to the existing literature, considerable attention is given to global visibility, link-state discovery, flow-rule positioning, and controller load unbalancing in complex and large-scale networks.

B. LESSONS LEARNED FROM OTHER APPROACHES INTEGRATED WITH SDN TO ADDRESS THE IOT FRAMEWORK'S MANAGEMENT CHALLENGES

1) Network Function Virtualization

SDNs have been widely deployed in the IoT environment, where they have been primarily used for flow optimization and related policies to manage IoT networks [215]. Virtualization in terms of networks, functions, and applications has also seen immense contributions in the recent past. To address IoT resource management problems, we studied both SDN and virtualization combination frameworks in the lit-

erature review and classified them into different IoT management solutions [216]. We notice that the SDN framework is limited to virtualizing the IP stack's network layer, where the traffic flow of the IoT network is configured. Therefore, in terms of implementation, the proposed solutions focus primarily on security solutions instead of combining other management issues in IoT networks

According to the existing literature, there are two ways to build an NFV/SDN-based architecture to solve IoT network management issues: one from the NFV side and the other from the SDN side. The NFV management and network orchestration (MANO) framework places various VNFs on the NFV side used in the SDN control plane, which provides multiple services to IoT networks such as security, load balancing, fault tolerance, etc. The SDN-side SDN controller in the NFV framework has its management strategies to solve the IoT management problem [50]. The majority of the proposed solution is distributed in nature, mainly focusing on fault tolerance and load balancing constraints with the help of SDN's flow tables, resource management access-lists, etc. Scalability is provided with the help of the NFV management framework. In terms of the NFV taxonomy, the proposed security solutions are mainly from the NFV side in which different virtual security solutions are provided to the IoT networks in the form of VNFs.

2) Middleware-based SDN Solution

The middleware layer plays a significant role in integration with the SDN controller to manage the IoT networks. This layer reduces the SDN controller's workload and provides additional benefits to the control plane and the data plane. In the gathered literature, most of the proposed frameworks' middleware layers consist of a perception layer, access layer, and edge layer. The majority of solutions focus on load-balancing, fault-tolerance, and scalability where the proposed algorithms in the middleware layer include SBIs and NBIs for control and data planes. The authors discuss the scalable middleware solution for interoperability across heterogeneous devices that serve in various application domains, such as discovery protocols to manage IoT devices and context-aware IoT applications also focus on computation and security parameters in order to provide efficient solutions to address the IoT management challenges ,minimal effort toward efficient modules that manage security, energy issues in IoT networks can be noticed.

3) OpenFLow

OpenFlow is the first dominant SDN flow control protocol, which has already been the defacto standard for SDN controllers [217]. Communication between the control layer and the forwarding layer is achieved through the southbound interface, and OpenFlow is one of the widely used southbound APIs [218]. The existing literature suggested that researchers put efforts to improve the management issues of IoT with the help of OpenFlow Southbound API. Still, we noticed that most of the implementation was done to manage load balancing issues in IoT, and a majority of the proposed frameworks lacked implementation details.

4) Blockchain

Blockchain technology is conceptually fundamentally different from SDN, as with a blockchain, information is decentralized in a P2P network and the need for a trusted third party is removed. In regards of accessibility of transactions, blockchains are classified as public, private, or consortium. In a public blockchain, all nodes take part in the consensus process and review the transaction data. On the contrary, in private and consortium blockchains, transaction accessibility is typically granted and revoked based on a centralized agency judgment. Only a small number of pre-approved nodes are involved in the consensus process. SDN breaks the vertical integration of the data and controls planes and passes the network's control logic to an SDN controller called a centralized entity [219]. SDN frameworks themselves have itself has some security limitations such as single point failure, improper network rule insertion, DDOS, etc., that may affect the performance of the IoT network. The combination of blockchain and SDN has the potential to manage IoT network resource management issues [220]. In the literature review, we find that security, scalability, decentralization, and traceability are the main features of blockchain technology that can assist an SDN-based framework in dealing with various challenges. Moreover, latency remains a constant challenge in blockchain-based SDN solutions in IoT networks. The majority of the related papers suggested that load balancing, fault tolerance, and energy management in IoT networks can be achieved with the help of blockchain-based smart-contracts.

X. OPEN RESEARCH CHALLENGES AND FUTURE OPPORTUNITIES

This section focuses on the SDIoT management framework's active research areas and open research problems connected to the defined taxonomy i,e., NFV, Middleware Openflow adaptation and Blockchain to address IoT management challenges.

A. NETWORK FUNCTION VIRTUALIZATION

In SDN-based frameworks for IoT networks, NFV is critical for properly handling data traffic and meeting resource management framework requirements. More research towards context-aware NFV employing AI to govern IoT frameworks is necessary [221]. During the last few years, the NFV based SD-IoT management solution has developed context-aware learning tools and systems. The most prevalent solutions are rule-based, logic-based, ontology-based, supervised, unsupervised, and reinforcement algorithms to improve performance [222]. It is possible to utilize a combination of hybrid machine learning approaches, such as rule-based and ensemble-based algorithms, to provide a better management framework and more advanced reasoning capabilities to address the management challenges [223]. Connecting each IoT

device to a power source is not always possible. IoT devices must be energy efficient in order to smooth the running of IoT network [224]. In the future, there is a need for power-hungry IoT devices with NFV-based architecture based on to save energy while maintaining QoS standards. Security challenges related to container-based virtualization technologies are also a popular topic. However, we can still ensure the security of container-based architectures by running them on top of VMs by using an adaptive approach [61].

B. MIDDLEWARE-BASED SOLUTIONS

In combination with edge computing, the emergence of the IoT has recently opened up several possibilities for new applications [225]. A common challenge is providing a persistent infrastructure, i.e., a service capable of continuously sustaining a high-efficiency level, facing potential failures, etc. In the future, there is a need for a middleware solution that works as a lightweight, adaptive engine in SDN-based frameworks to manage IoT resource management issues [224]. An in-depth investigation is needed to understand how centrally controlled IoT networks are managed via SDN-based IoT frameworks and how they can recover from faults, manage the sensor nodes' energy more efficiently, balance traffic within the network, and provide security to the network and its applications. Making the network status available to the SDN controller can help provide security services such as attack mitigation, privacy, lightweight key management, etc., for IoT networks [226]. Combining the proliferation of cloud-based network services as a middleware solution with SDN for solving IoT network management problems has created new challenges in cloud service selection, and ranking [227]. Because of the wide range of cloud services available, there is a need for IoT networks to select carefully the one that would suit their needs best and adjust to their circumstances accordingly. Because of the intelligent capabilities of network slicing and edge computing. Edge applications must have adjustability, dynamism, usability, flexibility, interoperability, and compatibility with other technologies are required [60].

C. OPENFLOW ADAPTATIONS

The combination of network programmability and IoT comes with new issues for IoT networks [228]. The most emphasized subject includes enforcing dynamic open-flow rules and procedures for various resource and security management issues, i.e., user authentication, software reliability, threat detection, lack of regular patches and updates, untrustworthy communication, and data privacy concerns [53]. Dedicated hardware appliances are replaced by programs running on virtual network functions (VNFs) that need intense packet processing under the network function virtualization paradigm. In the future, there is a need for VNF-based adaptive programmable rules-based distributed SDN switches managing mechanisms for load balancing, energy efficiency, data plane scalability, and traffic flow QoS requirements in IoT, in addition to Open Flow heterogeneous switches resources [229] Mobile nodes are the most common IoT devices that require mobility management protocols to deliver transparent services to users without delays or disconnections. Packet loss, end-to-end delay, increased handover latency, increased signalling costs, and power consumption is just a few of the concerns and problems that affect communication between mobile nodes in a mobile IP capable network. In future, there is a need for an AI-based adaptive protocol suite that handles the mobility management of IoT devices [230].

D. BLOCKCHAIN-BASED SOLUTIONS

Dynamic interoperability and protocol standardisation will be required in the future, posing further hurdles in addressing IoT device management issues in the smart city [60]. To achieve full Interoperability (i.e., from data to policy interoperability) and integration with heterogeneous IoT systems, the adoption of Blockchain will be the key that helps to overcome these challenges in IoT with the help of federated learning.Blockchain has been viewed as a viable fabric for a secure, decentralized IoT edge in recent years due to its inherent qualities of fault tolerance, transparency, and enforcement of service level agreements through smart contracts [59]. Despite their advantages, blockchains confront several challenges. One of the highlighted challenges is in the on-demand decentralized horizontal scaling of an IoT-based smart city networks.Blockchain-based decentralized security frameworks for IoT networks that works adaptively and dynamically to adopt multiple security solutions will highly be required in the near future [58].

XI. CONCLUSION

The IoT paradigm presents a future of computing that is rapidly gaining traction in our lives as a means of improving the quality of life by connecting a range of intelligent devices, technologies, services, and applications. However, there are several challenges within the IoT network management frameworks that require novel solutions. These challenges revolve around the fragile nature of IoT devices in terms of faults; failure in the wake of higher traffic load; security weaknesses; the lack of energy efficiency; and scalability. IoT devices are heterogeneous and resource-constrained. The operation of these diverse IoT devices requires specialized network behavior and services such as security, efficient energy management, load management module, etc., that is also overhead to the IoT networks. SDN, with its novel approaches to network management along with its latest developments within the realm of IoT offers promising solutions. SDN provides global visibility of the network state and logically centralized control of resources, which can be physically distributed if required, through programmable APIs from a central vintage point. Thus, SDN facilitates novel techniques for network management. Therefore, huge research efforts are dedicated to developing SDN-based IoT management frameworks.

This article presents a detailed overview of the state-of-the-art of important SDN-based IoT management frameworks. These frameworks are discussed in terms of four key trends: 1) NFV-based frameworks, 2) Middleware-based frameworks, 3) OpenFlow-based frameworks, and 4) Blockchain-based frameworks. All the proposed architectures discussed in this article are utilizing the reconfiguration capabilities of SDNs, which is fundamental to existing and future IoT systems. The main theme in these four dimensions is to improve fault tolerance, energy and security management, load balancing, and improving scalability. Albeit, SDN lays the foundation for robust management solutions, AI-based approaches in conjunction with SDN are still lacking to embed intelligent decision-making during uncertain situations. Blockchain, IoT, and AI are innovations that can promise benefits in security, transparency, immutability, privacy, and business process automation in IoT networks. However, when blockchain, IoT, and AI are combined into an SDN framework to manage IoT networks, the benefits of these technologies can even be higher. In the future, we envision that with the help of AI, adaptive resource management frameworks for IoT networks will be introduced that will also include blockchain-based SDN frameworks. Moreover, the envisioned deployment of IoT on a wide scale would reveal further practical challenges since most of the existing research is either in constrained and lab environments or based on theoretical evaluations. The state-of-the-art research work identified in this article suggests that the dynamism provided by SDN can help reconfigure or update and upgrade the IoT network at run-time to solve emerging challenges.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys Tutorials, 17(4):2347–2376, 2015.

[2] Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, Mika Ylianttila, Timo Koskela, Timo Braysy, Antti Anttonen, Vesa Pentikinen, Juha-Pekka Soininen, and Jyrki Huusko. Towards gadget-free Internet services: A roadmap of the naked world. Telematics and Informatics, 35(1):82–92, 2018.

[3] Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, and Ibrar Yaqoob. Big IoT data analytics: architecture, opportunities, and open research challenges. IEEE Access, 5:5247–5261, 2017.

[4] Tie Qiu, Ning Chen, Keqiu Li, Mohammed Atiquzzaman, and Wenbing Zhao. How can heterogeneous Internet of things build our future: A survey. IEEE Communications Surveys & Tutorials, 20(3):2011–2027, 2018.

[5] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7):1645–1660, 2013.

[6] Bin Guo, Daqing Zhang, Zhu Wang, Zhiwen Yu, and Xingshe Zhou. Opportunistic IoT: Exploring the harmonious interaction between human and the Internet of things. Journal of Network and Computer Applications, 36(6):1531–1539, 2013.

[7] T. Park, N. Abuzainab, and W. Saad. Learning how to communicate in the Internet of things: Finite resources and heterogeneity. IEEE Access, 4:7063–7073, 2016.

[8] Intidhar Bedhief, Meriem Kassar, and Taoufik Aguili. SDN-based architecture challenging the IoT heterogeneity. In 2016 3rd Smart Cloud Networks & Systems (SCNS), pages 1–3. IEEE, 2016.

[9] Laith Farhan, Sinan T Shukur, Ali E Alissa, Mohmad Alrweg, Umar Raza, and Rupak Kharel. A survey on the challenges and opportunities of the Internet of things (IoT). In 2017 Eleventh International Conference on Sensing Technology (ICST), pages 1–5. IEEE, 2017.

[10] Pritish Mishra, Deepak Puthal, Mayank Tiwary, and Saraju P Mohanty. Software defined IoT systems: Properties, state of the art, and future research. IEEE Wireless Communications, 26(6):64–71, 2019.

[11] Hlabishi I Kobo, Adnan M Abu-Mahfouz, and Gerhard P Hancke. A survey on software-defined wireless sensor networks: Challenges and design requirements. IEEE Access, 5:1872–1899, 2017.

[12] Muhammad Ali Hassan, Quoc-Tuan Vien, and Mahdi Aiash. Software defined networking for wireless sensor networks: a survey. Advances in Wireless Communications and Networks, 3(2):10–22, 2017.

[13] Hai Huang, Jiping Zhu, and Lei Zhang. An sdn_based management framework for IoT devices. ZTE communications, 2014.

[14] Olivier Flauzac, Carlos González, Abdelhak Hachani, and Florent Nolot. SDN based architecture for IoT and improvement of the security. In 2015 ieee 29th international conference on advanced information networking and applications workshops, pages 688–693. IEEE, 2015.

[15] Yifei Yuan, Dong Lin, Rajeev Alur, and Boon Thau Loo. Scenario-based programming for SDN policies. In Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, pages 1–13, 2015.

[16] Zhijing Qin, Grit Denker, Carlo Giannelli, Paolo Bellavista, and Nalini Venkatasubramanian. A software defined networking architecture for the Internet-of-things. In 2014 IEEE network operations and management symposium (NOMS), pages 1–9. IEEE, 2014.

[17] Peng Hu. A system architecture for software-defined industrial Internet of things. In 2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), pages 1–5. IEEE, 2015.

[18] Vishwapathi Rao Tadinada. Software defined networking: Redefining the future of Internet in IoT and cloud era. In 2014 2nd International Conference on Future Internet of Things and Cloud (FiCloud), pages 296–301. IEEE Computer Society, 2014.

[19] Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. SDN security: A survey. In 2013 IEEE SDN For Future Networks and Services (SDN4FNS), pages 1–7. IEEE, 2013.

[20] Nikos Bizanis and Fernando A Kuipers. SDN and virtualization solutions for the Internet of things: A survey. IEEE Access, 4:5591–5606, 2016.

[21] Manuel Díaz, Cristian Martín, and Bartolomé Rubio. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. Journal of Network and Computer applications, 67:99–117, 2016.

[22] Qiang Duan, Nirwan Ansari, and Mehmet Toy. Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks. IEEE Network, 30(5):10–16, 2016.

[23] Christos Bouras, Anastasia Kollia, and Andreas Papazois. SDN & NFV in 5G: Advancements and challenges. In 2017 20th Conference on innovations in clouds, internet and networks (ICIN), pages 107–111. IEEE, 2017.

[24] Sahrish Khan Tayyaba, Munam Ali Shah, Omair Ahmad Khan, and Abdul Wahab Ahmed. Software defined network (SDN) based Internet of things (IoT) a road ahead. In Proceedings of the International Conference on Future Networks and Distributed Systems, pages 1–8, 2017.

[25] Samaresh Bera, Sudip Misra, and Athanasios V Vasilakos. Software-defined networking for Internet of things: A survey. IEEE Internet of Things Journal, 4(6):1994–2008, 2017.

[26] Xinli Huang, Shang Cheng, Kun Cao, Peijin Cong, Tongquan Wei, and Shiyan Hu. A survey of deployment solutions and optimization strategies for hybrid SDN networks. IEEE Communications Surveys & Tutorials, 21(2):1483–1507, 2018.

[27] Heng Zhang, Zhiping Cai, Qiang Liu, Qingjun Xiao, Yangyang Li, and Chak Fone Cheang. A survey on security-aware measurement in SDN. Security and Communication Networks, 2018, 2018.

[28] Mohamed Amine Ferrag, Makhlouf Derdour, Mithun Mukherjee, Abdelouahid Derhab, Leandros Maglaras, and Helge Janicke. Blockchain technologies for the Internet of things: Research issues and challenges. IEEE Internet of Things Journal, 6(2):2188–2204, 2018.

[29] Weichao Gao, William G Hatcher, and Wei Yu. A survey of blockchain: Techniques, applications, and challenges. In 2018 27th international conference on computer communication and networks (ICCCN), pages 1–11. IEEE, 2018.

[30] Ivan Farris, Tarik Taleb, Yacine Khettab, and Jaeseung Song. A survey on emerging SDN and NFV security mechanisms for IoT systems. IEEE Communications Surveys & Tutorials, 21(1):812–837, 2018.

[31] Hamza Zemrane, Youssef Baddi, and Abderrahim Hasbi. SDN-based solutions to improve IoT: survey. In 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), pages 588–593. IEEE, 2018.

[32] Renuga Kanagavelu and Khin Mi Mi Aung. A survey on SDN based security in Internet of things. In Future of Information and Communication Conference, pages 563–577. Springer, 2018.

[33] Muhammad Burhan, Rana Asif Rehman, Bilal Khan, and Byung-Seo Kim. IoT elements, layered architectures and security issues: A comprehensive survey. Sensors, 18(9):2796, 2018.

[34] Muhammad Salek Ali, Massimo Vecchio, Miguel Pincheira, Koustabh Dolui, Fabio Antonelli, and Mubashir Husain Rehmani. Applications of blockchains in the Internet of things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 21(2):1676–1717, 2018.

[35] Fabiola Hazel Pohrmen, Rohit Kumar Das, and Goutam Saha. Blockchain-based security aspects in heterogeneous Internet-of-things networks: A survey. Transactions on Emerging Telecommunications Technologies, 30(10):e3741, 2019.

[36] Xu Wang, Xuan Zha, Wei Ni, Ren Ping Liu, Y Jay Guo, Xinxin Niu, and Kangfeng Zheng. Survey on blockchain for Internet of things. Computer Communications, 136:10–29, 2019.

[37] Mingli Wu, Kun Wang, Xiaoqin Cai, Song Guo, Minyi Guo, and Chunming Rong. A comprehensive survey of blockchain: From theory to IoT applications and beyond. IEEE Internet of Things Journal, 6(5):8114–8154, 2019.

[38] Mohammed Alsaeedi, Mohd Murtadha Mohamad, and Anas A Al-Roubaiey. Toward adaptive and scalable OpenFlow-SDN flow control: A survey. IEEE Access, 7:107346–107379, 2019.

[39] Anish Jindal, Gagangeet Singh Aujla, and Neeraj Kumar. SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment. Computer Networks, 153:36–48, 2019.

[40] Noshina Tariq, Muhammad Asim, Feras Al-Obeidat, Muhammad Zubair Farooqi, Thar Baker, Mohammad Hammoudeh, and Ibrahim Ghafir. The security of big data in fog-enabled IoT applications including blockchain: A survey. Sensors, 19(8):1788, 2019.

[41] Rajat Chaudhary, Anish Jindal, Gagangeet Singh Aujla, Shubhani Aggarwal, Neeraj Kumar, and Kim-Kwang Raymond Choo. BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system. Computers & Security, 85:288–299, 2019.

[42] Anteneh A Gebremariam, Muhammad Usman, and Marwa Qaraqe. Applications of artificial intelligence and machine learning in the area of SDN and NFV: A survey. In 2019 16th International Multi-Conference on Systems, Signals & Devices (SSD), pages 545–549. IEEE, 2019.

[43] PJ Beslin Pajila and E Golden Julie. Detection of DDoS attack using SDN in IoT: a survey. In Intelligent Communication Technologies and Virtual Mobile Networks, pages 438–452. Springer, 2019.

[44] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. LSB: A lightweight scalable blockchain for IoT security and anonymity. Journal of Parallel and Distributed Computing, 134:180–197, 2019.

[45] Bhabendu Kumar Mohanta, Debasish Jena, Utkalika Satapathy, and Srikanta Patnaik. Survey on IoT security: challenges and solution using machine learning, artificial intelligence and blockchain technology. Internet of Things, page 100227, 2020.

[46] Jiejun Hu, Martin Reed, Nikolaos Thomos, Mays F AI-Naday, and Kun Yang. Securing SDN controlled IoT networks through edge-blockchain. IEEE Internet of Things Journal, 2020.

[47] Iqbal Alam, Kashif Sharif, Fan Li, Zohaib Latif, MM Karim, Sujit Biswas, Boubakr Nour, and Yu Wang. A survey of network virtualization techniques for internet of things using SDN and NFV. ACM Computing Surveys (CSUR), 53(2):1–40, 2020.

[48] Congfeng Jiang, Tiantian Fan, Honghao Gao, Weisong Shi, Liangkai Liu, Christophe Cerin, and Jian Wan. Energy aware edge computing: A survey. Computer Communications, 151:556–580, 2020.

[49] Wajid Rafique, Lianyong Qi, Ibrar Yaqoob, Muhammad Imran, Raihan Ur Rasool, and Wanchun Dou. Complementing iot services through software defined networking and edge computing: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(3):1761–1804, 2020.

[50] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. Computer Networks, 167:106984, 2020.

[51] Dipon Saha, Meysam Shojaee, Michael Baddeley, and Israat Haque. An energy-aware SDN/NFV architecture for the Internet of things. In 2020 IFIP Networking Conference (Networking), pages 604–608. IEEE, 2020.

[52] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M Hadi Amini. A survey on federated learning for resource-constrained iot devices. IEEE Internet of Things Journal, 2021.

[53] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. Future Generation Computer Systems, 115:619–640, 2021.

[54] Partha Pratim Ray and Neeraj Kumar. SDN/NFV architectures for edge-cloud oriented IoT: A systematic review. Computer Communications, 2021.

[55] Md Ashraf Uddin, Andrew Stranieri, Iqbal Gondal, and Venki Balasubramanian. A survey on the adoption of blockchain in IoT: Challenges and solutions. Blockchain: Research and Applications, page 100006, 2021.

[56] Suhail Ahmad and Ajaz Hussain Mir. Scalability, consistency, reliability and security in SDN controllers: A survey of diverse SDN controllers. Journal of Network and Systems Management, 29(1):1–59, 2021.

[57] Sajad Khorsandroo, Adrián Gallego Sánchez, Ali Saman Tosun, José Manuel Arco Rodríguez, and Roberto Doriguzzi-Corin. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. Computer Networks, page 107981, 2021.

[58] Nada Alasbali, Saaidal Razalli Bin Azzuhri, Rosli Bin Salleh, Miss Laiha Mat Kiah, Ahmad Aliff AS Shariffuddin, Nik Muhammad Izwan bin Nik Mohd Kamel, and Leila Ismail. Rules of smart iot networks within smart cities towards blockchain standardization. Mobile Information Systems, 2022, 2022.

[59] Imran Ahmed, Yulan Zhang, Gwanggil Jeon, Wenmin Lin, Mohammad R Khosravi, and Lianyong Qi. A blockchain-and artificial intelligence-enabled smart iot framework for sustainable city. International Journal of Intelligent Systems, 2022.

[60] Bimal Ghimire and Danda B Rawat. Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. IEEE Internet of Things Journal, 2022.

[61] Zawar Shah, Imdad Ullah, Huiling Li, Andrew Levula, and Khawar Khurshid. Blockchain based solutions to mitigate distributed denial of service (ddos) attacks in the internet of things (iot): A survey. Sensors, 22(3):1094, 2022.

[62] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications surveys & tutorials, 16(3):1617–1634, 2014.

[63] Ola Salman, Imad Elhajj, Ali Chehab, and Ayman Kayssi. IoT survey: An SDN and fog computing perspective. Computer Networks, 143:221–246, 2018.

[64] Musa Ndiaye, Gerhard P Hancke, and Adnan M Abu-Mahfouz. Software defined networking for improved wireless sensor network management: A survey. Sensors, 17(5):1031, 2017.

[65] Waseem Iqbal, Haider Abbas, Mahmoud Daneshmand, Bilal Rauf, and Yawar Abbas Bangash. An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. IEEE Internet of Things Journal, 7(10):10250–10276, 2020.

[66] Muneer Bani Yassein, Shadi Aljawarneh, Mohammad Al-Rousan, Wail Mardini, and Wesam Al-Rashdan. Combined software-defined network (SDN) and Internet of things (IoT). In 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), pages 1–6. IEEE, 2017.

[67] Jue Chen, Jinbang Chen, Fei Xu, Min Yin, and Wei Zhang. When software defined networks meet fault tolerance: A survey. In International conference on algorithms and architectures for parallel processing, pages 351–368. Springer, 2015.

[68] Yinbo Yu, Xing Li, Xue Leng, Libin Song, Kai Bu, Yan Chen, Jianfeng Yang, Liang Zhang, Kang Cheng, and Xin Xiao. Fault management in software-defined networking: A survey. IEEE Communications Surveys & Tutorials, 21(1):349–392, 2018.

[69] Tao Wang, Fangming Liu, and Hong Xu. An efficient online algorithm for dynamic SDN controller assignment in data center networks. IEEE/ACM Transactions on Networking, 25(5):2788–2801, 2017.

[70] Sana Moin, Ahmad Karim, Kalsoom Safdar, Iqra Iqbal, Zanab Safdar, V Vijayakumar, KT Ahmed, and SA Abid. GREEN SDN – an enhanced

paradigm of SDN: Review, taxonomy, and future directions. Concurrency and Computation: Practice and Experience, 32(21):e5086, 2020.

[71] Ikhlasse Hamzaoui, Benjamin Duthil, Vincent Courboulay, and Hicham Medromi. A survey on the current challenges of energy-efficient cloud resources management. SN Computer Science, 1(2):1–28, 2020.

[72] Kashif Inayat and Seong Oun Hwang. Load balancing in decentralized smart grid trade system using blockchain. Journal of Intelligent & Fuzzy Systems, 35(6):5901–5911, 2018.

[73] Zhaogang Shu, Jiafu Wan, Jiaxiang Lin, Shiyong Wang, Di Li, Seungmin Rho, and Changcai Yang. Traffic engineering in software-defined networking: Measurement and management. IEEE access, 4:3246–3256, 2016.

[74] Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov. Security in software defined networks: A survey. IEEE Communications Surveys Tutorials, 17(4):2317–2346, 2015.

[75] Jalal Bhayo, Sufian Hameed, and Syed Attique Shah. An efficient counter-basedDDoS attack detection framework leveraging software defined IoT (SD-IoT). IEEE Access, 8:221612–221631, 2020.

[76] Narmeen Zakaria Bawany, Jawwad A Shamsi, and Khaled Salah. DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arabian Journal for Science and Engineering, 42(2):425–441, 2017.

[77] Mehrnoosh Monshizadeh, Vikramajeet Khatri, and Raimo Kantola. An adaptive detection and prevention architecture for unsafe traffic in SDN enabled mobile networks. In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pages 883–884. IEEE, 2017.

[78] Sujit Biswas, Kashif Sharif, Fan Li, Boubakr Nour, and Yu Wang. A scalable blockchain framework for secure transactions in IoT. IEEE Internet of Things Journal, 6(3):4650–4659, 2018.

[79] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – a systematic literature review. Information and software technology, 51(1):7–15, 2009.

[80] Barbara Kitchenham. Procedures for performing systematic reviews. Keele, UK, Keele University, 33(2004):1–26, 2004.

[81] Myung-Ki Shin, Ki-Hyuk Nam, and Hyoung-Jun Kim. Software-defined networking (SDN): A reference architecture and open APIs. In 2012 International Conference on ICT Convergence (ICTC), pages 360–361. IEEE, 2012.

[82] Hesham Mekky, Fang Hao, Sarit Mukherjee, Zhi-Li Zhang, and TV Lakshman. Application-aware data plane processing in SDN. In Proceedings of the third workshop on Hot topics in software defined networking, pages 13–18, 2014.

[83] Akihiro Nakao. Software-defined data plane enhancing SDN and NFV. IEICE Transactions on Communications, 98(1):12–19, 2015.

[84] Aparna Kumari, Rajesh Gupta, Sudeep Tanwar, and Neeraj Kumar. A taxonomy of blockchain-enabled softwarization for secure UAV network. Computer Communications, 161:304–323, 2020.

[85] Kevin Phemius, Mathieu Bouet, and Jérémie Leguay. Disco: Distributed multi-domain SDN controllers. In 2014 IEEE Network Operations and Management Symposium (NOMS), pages 1–4. IEEE, 2014.

[86] V Thirupathi, CH Sandeep, Naresh Kumar, and Pramod Kumar. A comprehensive review on SDN architecture, applications and major benifits of SDN. International Journal of Advanced Science and Technology, 28(20):607–614, 2019.

[87] Shie-Yuan Wang, Hung-Wei Chiu, and Chih-Liang Chou. Comparisons of SDN OpenFlow controllers over EstiNet: Ryu vs. NOX. Proceedings of the ICN 2015, page 256, 2015.

[88] Marco Canini, Daniele Venzano, Peter Perešíni, Dejan Kostić, and Jennifer Rexford. A NICE way to test OpenFlow applications. In Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12), pages 127–140, 2012.

[89] Richard Skowyra, Andrei Lapets, Azer Bestavros, and Assaf Kfoury. A verification platform for SDN-enabled applications. In 2014 IEEE International Conference on Cloud Engineering, pages 337–342. IEEE, 2014.

[90] Thomas Ball, Nikolaj Bjørner, Aaron Gember, Shachar Itzhaky, Aleksandr Karbyshev, Mooly Sagiv, Michael Schapira, and Asaf Valadarsky. VeriCon: towards verifying controller programs in software-defined networks. In Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 282–293, 2014.

[91] Yustus Eko Oktian, SangGon Lee, HoonJae Lee, and JunHuy Lam. Distributed SDN controller system: A survey on design choice. computer networks, 121:100–111, 2017.

[92] Sanjeev Singh and Rakesh Kumar Jha. A survey on software defined networking: Architecture for next generation network. Journal of Network and Systems Management, 25(2):321–374, 2017.

[93] Sakir Sezer, Sandra Scott-Hayward, Pushpinder Kaur Chouhan, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Marc Miller, and Navneet Rao. Are we ready for SDN? implementation challenges for software-defined networks. IEEE Communications Magazine, 51(7):36–43, 2013.

[94] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2):69–74, 2008.

[95] Adrian Lara, Anisha Kolasani, and Byrav Ramamurthy. Network innovation using OpenFlow: A survey. IEEE communications surveys & tutorials, 16(1):493–512, 2013.

[96] Ian F Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, and Wu Chou. A roadmap for traffic engineering in SDN-OpenFlow networks. Computer Networks, 71:1–30, 2014.

[97] Belema Agborubere and Erika Sanchez-Velazquez. OpenFlow communications and TLS security in software-defined networks. In 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pages 560–566. IEEE, 2017.

[98] Avri Doria, Jamal Hadi Salim, Robert Haas, Hormuzd M Khosravi, Weiming Wang, Ligang Dong, Ram Gopal, and Joel M Halpern. Forwarding and control element separation (ForCES) protocol specification. RFC, 5810:1–124, 2010.

[99] Arpita Prajapati, Achyut Sakadasariya, and Jitisha Patel. Software defined network: Future of networking. In 2018 2nd International Conference on Inventive Systems and Control (ICISC), pages 1351–1354. IEEE, 2018.

[100] Manish Paliwal, Deepti Shrimankar, and Omprakash Tembhurne. Controllers in SDN: A review report. IEEE Access, 6:36256–36270, 2018.

[101] Mohammad Nowsin Amin Sheikh. SDN-based approach to evaluate the best controller: Internal controller NOX and external controllers POX, ONOS, RYU. Global Journal of Computer Science and Technology, 2019.

[102] Zuhran Khan Khattak, Muhammad Awais, and Adnan Iqbal. Performance evaluation of OpenDaylight SDN controller. In 2014 20th IEEE international conference on parallel and distributed systems (ICPADS), pages 671–676. IEEE, 2014.

[103] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, et al. ONOS: towards an open, distributed SDN OS. In Proceedings of the third workshop on Hot topics in software defined networking, pages 1–6, 2014.

[104] Sungmin Hong, Lei Xu, Haopei Wang, and Guofei Gu. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In Ndss, volume 15, pages 8–11, 2015.

[105] Ibrahim Afolabi, Tarik Taleb, Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. IEEE Communications Surveys & Tutorials, 20(3):2429–2453, 2018.

[106] Corrado Rametta and Giovanni Schembra. Designing a softwarized network deployed on a fleet of drones for rural zone monitoring. Future Internet, 9(1):8, 2017.

[107] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The road to SDN: an intellectual history of programmable networks. ACM SIGCOMM Computer Communication Review, 44(2):87–98, 2014.

[108] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. NOX: towards an operating system for networks. ACM SIGCOMM Computer Communication Review, 38(3):105–110, 2008.

[109] Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer. A survey of security in software defined networks. IEEE Communications Surveys & Tutorials, 18(1):623–654, 2015.

[110] Ali Akbar Neghabi, Nima Jafari Navimipour, Mehdi Hosseinzadeh, and Ali Rezaee. Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. IEEE Access, 6:14159–14178, 2018.

[111] Isravel Deva Priya and Salaja Silas. A survey on research challenges and applications in empowering the SDN-based Internet of things. Advances in Big Data and Cloud Computing, pages 457–467, 2019.

[112] Mehmet Fatih Tuysuz, Zekiye Kubra Ankarali, and Didem Gözüpek. A survey on energy efficiency in software defined networks. Computer Networks, 113:188–204, 2017.

[113] Yousaf Bin Zikria, Sung Won Kim, Oliver Hahm, Muhammad Khalil Afzal, and Mohammed Y Aalsalem. Internet of things (IoT) operating systems management: opportunities, challenges, and solution, 2019.

[114] Mohammad Asad Abbasi, Zulfiqar A Memon, Jamshed Memon, Tahir Q Syed, and Rabah Alshboul. Addressing the future data management challenges in IoT: A proposed framework. International Journal of Advanced Computer Science and Applications, 8(5):197–207, 2017.

[115] Theodoros Anagnostopoulos, Arkady Zaslavsky, Kostas Kolomvatsos, Alexey Medvedev, Pouria Amirian, Jeremy Morley, and Stathes Hadjieftymiades. Challenges and opportunities of waste management in IoT-enabled smart cities: A survey. IEEE Transactions on Sustainable Computing, 2(3):275–289, 2017.

[116] Kour Tejasvit. Challenges in integrating wireless sensor networks into the Internet. International Journal of Engineering and Management Sciences, 5(1):7–11, 2014.

[117] Rashid Amin, Martin Reisslein, and Nadir Shah. Hybrid SDN networks: A survey of existing approaches. IEEE Communications Surveys & Tutorials, 20(4):3259–3306, 2018.

[118] Hanane Lamaazi, Nabil Benamar, Antonio J Jara, Latif Ladid, and Driss El Ouadghiri. Challenges of the Internet of things: IPv6 and network management. In 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pages 328–333. IEEE, 2014.

[119] Arjun P Athreya and Patrick Tague. Network self-organization in the Internet of things. In 2013 IEEE international workshop of internet-of-things networking and control (IoT-NC), pages 25–33. IEEE, 2013.

[120] Alem Čolaković and Mesud Hadžialić. Internet of things (IoT): A review of enabling technologies, challenges, and open research issues. Computer Networks, 144:17–39, 2018.

[121] Mohamed Abomhara and Geir M Køien. Security and privacy in the Internet of things: Current status and open issues. In 2014 international conference on privacy and security in mobile systems (PRISMS), pages 1–8. IEEE, 2014.

[122] Ihsan Ali, Abdelmuttlib Ibrahim Abdalla Ahmed, Ahmad Almogren, Muhammad Ahsan Raza, Syed Attique Shah, Anwar Khan, and Abdullah Gani. Systematic literature review on IoT-based botnet attack. IEEE Access, 2020.

[123] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. Security, privacy and trust in Internet of things: The road ahead. Computer networks, 76:146–164, 2015.

[124] Zheng Yan, Peng Zhang, and Athanasios V Vasilakos. A survey on trust management for Internet of things. Journal of network and computer applications, 42:120–134, 2014.

[125] NEW SOUTH WALES. Securing networks using software defined networking: a survey. ZTE communications, 2013.

[126] Mahyar Tourchi Moghaddam and Henry Muccini. Fault-tolerant IoT. In International Workshop on Software Engineering for Resilient Systems, pages 67–84. Springer, 2019.

[127] Asad Javed, Keijo Heljanko, Andrea Buda, and Kary Främling. CEFIoT: A fault-tolerant IoT architecture for edge and cloud. In 2018 IEEE 4th world forum on internet of things (WF-IoT), pages 813–818. IEEE, 2018.

[128] Lilia Paradis and Qi Han. A survey of fault management in wireless sensor networks. Journal of Network and systems management, 15(2):171–190, 2007.

[129] Hafedh Trigui, Rick Cuthill, and Richard Glenn Kusyk. Dynamic load balancing, July 30 2013. US Patent 8,498,207.

[130] Fadi M Al-Turjman, Ashraf E Al-Fagih, Waleed M Alsalih, and Hossam S Hassanein. A delay-tolerant framework for integrated RSNs in IoT. Computer Communications, 36(9):998–1010, 2013.

[131] Shih-Yeh Chen, Chin-Feng Lai, Yueh-Min Huang, and Yu-Lin Jeng. Intelligent home-appliance recognition over IoT cloud network. In 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pages 639–643. IEEE, 2013.

[132] László Lengyel, Péter Ekler, Tamás Ujj, Tamás Balogh, and Hassan Charaf. SensorHUB: An IoT driver framework for supporting sensor networks and data analysis. International Journal of Distributed Sensor Networks, 11(7):454379, 2015.

[133] Dipak Wajgi and Nileshsingh V Thakur. Load balancing algorithms in wireless sensor network: a survey. International Journal of Computer Networks and Wireless Communications (IJCNWC), 2:456–460, 2012.

[134] Bob Duncan, Andreas Happe, and Alfred Bratterud. Enterprise IoT security and scalability: how unikernels can improve the status quo. In 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), pages 292–297. IEEE, 2016.

[135] Kyriakos Georgiou, Samuel Xavier-de Souza, and Kerstin Eder. The IoT energy challenge: A software perspective. IEEE Embedded Systems Letters, 10(3):53–56, 2017.

[136] Danda B Rawat and Swetha R Reddy. Software defined networking architecture, security and energy efficiency: A survey. IEEE Communications Surveys & Tutorials, 19(1):325–346, 2016.

[137] Mike Ojo, Davide Adami, and Stefano Giordano. A sdn-IoT architecture with NFV implementation. In 2016 IEEE Globecom Workshops (GC Wkshps), pages 1–6. IEEE, 2016.

[138] Van-Giang Nguyen, Anna Brunstrom, Karl-Johan Grinnemo, and Javid Taheri. SDN/NFV-based mobile packet core network architectures: A survey. IEEE Communications Surveys & Tutorials, 19(3):1567–1602, 2017.

[139] Ramon Ferrús, Harilaos Koumaras, Oriol Sallent, George Agapiou, Tinku Rasheed, M-A Kourtis, C Boustie, Patrick Gélard, and Toufik Ahmed. SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges. Physical Communication, 18:95–112, 2016.

[140] Matthew Monaco, Oliver Michel, and Eric Keller. Applying operating system principles to SDN controller design. In Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks, pages 1–7, 2013.

[141] Karamjeet Kaur, Veenu Mangat, and Krishan Kumar. A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture. Computer Science Review, 38:100298, 2020.

[142] Juliver Gil Herrera and Juan Felipe Botero. Resource allocation in NFV: A comprehensive survey. IEEE Transactions on Network and Service Management, 13(3):518–532, 2016.

[143] Nathalie Omnes, Marc Bouillon, Gael Fromentoux, and Olivier Le Grand. A programmable and virtualized network & IT infrastructure for the Internet of things: How can NFV & SDN help for facing the upcoming challenges. In 2015 18th International Conference on Intelligence in Next Generation Networks, pages 64–69. IEEE, 2015.

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

[145] Md Jahidul Islam, Md Mahin, Shanto Roy, Biplab Chandra Debnath, and Ayesha Khatun. Distblacknet: A distributed secure black SDN-IoT architecture with NFV implementation for smart cities. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), pages 1–6. IEEE, 2019.

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

[147] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N Asokan, and Ahmad-Reza Sadeghi. DIoT: A federated self-learning anomaly detection system for IoT. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pages 756–767. IEEE, 2019.

[148] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-BaIoT – network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3):12–22, 2018.

[149] Yehuda Afek, Anat Bremler-Barr, David Hay, Ran Goldschmidt, Lior Shafir, Gafnit Avraham, and Avraham Shalev. NFV-based IoT security for home networks using MUD. In NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, pages 1–9. IEEE, 2020.

[150] Ping Du, Pratama Putra, Shu Yamamoto, and Akihiro Nakao. A context-aware IoT architecture through software-defined data plane. In 2016 IEEE Region 10 Symposium (TENSYMP), pages 315–320. IEEE, 2016.

[151] Philippe Massonet, Laurent Deru, Amel Achour, Sebastien Dupont, Louis-Marie Croisez, Anna Levin, and Massimo Villari. Security in lightweight network function virtualisation for federated cloud and IoT. In 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), pages 148–154. IEEE, 2017.

[152] Changhao Zhang. Design and application of fog computing and Internet of things service platform for smart city. Future Generation Computer Systems, 112:630–640, 2020.

[153] Jose Costa-Requena, Jesús Llorente Santos, Vicent Ferrer Guasch, Kimmo Ahokas, Gopika Premsankar, Sakari Luukkainen, Oscar López Pérez, Mikel Uriarte Itzazelaia, Ijaz Ahmad, Madhusanka Liyanage, et al. SDN and NFV integration in generalized mobile network architecture. In 2015 European conference on networks and communications (EuCNC), pages 154–158. IEEE, 2015.

[154] Ahmed Dawoud, Seyed Shahristani, and Chun Raun. Deep learning and software-defined networks: Towards secure IoT architecture. Internet of Things, 3:82–89, 2018.

[155] Yuhong Li, Xiang Su, Jukka Riekki, Theo Kanter, and Rahim Rahmani. A SDN-based architecture for horizontal Internet of things services. In 2016 IEEE International Conference on Communications (ICC), pages 1–7. IEEE, 2016.

[156] Tanweer Alam. A middleware framework between mobility and IoT using IEEE 802.15.4e sensor networks. Jurnal Online Informatika, 4(2):90–94, 2020.

[157] Sharu Bansal and Dilip Kumar. IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication. International Journal of Wireless Information Networks, pages 1–25, 2020.

[158] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti, and Subhajit Dutta. Role of middleware for Internet of things: A study. International Journal of Computer Science and Engineering Survey, 2(3):94–105, 2011.

[159] Sergio D Castilho, Eduardo P Godoy, and Fadir Salmen. Implementing security and trust in IoT/M2M using middleware. In 2020 International Conference on Information Networking (ICOIN), pages 726–731. IEEE, 2020.

[160] Sukanya Bhowmik, Muhammad Adnan Tariq, Boris Koldehofe, Frank Dürr, Thomas Kohler, and Kurt Rothermel. High performance publish/subscribe middleware in software-defined networks. IEEE/ACM Transactions on Networking, 25(3):1501–1516, 2016.

[161] Aleksandar Antonić, Martina Marjanović, Krešimir Pripužić, and Ivana Podnar Žarko. A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of things. Future Generation Computer Systems, 56:607–622, 2016.

[162] Yali Wang, Yang Zhang, and Junliang Chen. SDNPS: A load-balanced topic-based publish/subscribe system in software-defined networking. Applied Sciences, 6(4):91, 2016.

[163] Akram Hakiri, Pascal Berthou, Aniruddha Gokhale, and Slim Abdellatif. Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications. IEEE communications magazine, 53(9):48–54, 2015.

[164] Yali Wang, Yang Zhang, and Junliang Chen. AnSDN-based publish/subscribe-enabled communication platform for IoT services. China communications, 15(1):95–106, 2018.

[165] Pedro F Moraes and Joberto SB Martins. A pub/sub SDN-integrated framework for IoT traffic orchestration. In Proceedings of the 3rd International Conference on Future Networks and Distributed Systems, pages 1–9, 2019.

[166] Yaser Jararweh, Mahmoud Al-Ayyoub, Elhadj Benkhelifa, Mladen Vouk, Andy Rindos, et al. SDIoT: a software defined based Internet of things framework. Journal of Ambient Intelligence and Humanized Computing, 6(4):453–461, 2015.

[167] Laura Galluccio, Sebastiano Milardo, Giacomo Morabito, and Sergio Palazzo. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In 2015 IEEE Conference on Computer Communications (INFOCOM), pages 513–521. IEEE, 2015.

[168] Martin Jacobsson and Charalampos Orfanidis. Using software-defined networking principles for wireless sensor networks. In SNCNW 2015, May 28–29, Karlstad, Sweden, 2015.

[169] Christian Jacquenet and Mohamed Boucadair. A software-defined approach to IoT networking. ZTE Communications, 1:012, 2016.

[170] Di Wu, Dmitri I Arkhipov, Eskindir Asmare, Zhijing Qin, and Julie A McCann. UbiFlow: Mobility management in urban-scale software defined IoT. In 2015 IEEE conference on computer communications (INFOCOM), pages 208–216. IEEE, 2015.

[171] Hakiri Akram and Aniruddha Gokhale. Rethinking the design of LR-WPAN IoT systems with software-defined networking. In 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS), pages 238–243. IEEE, 2016.

[172] Jiafu Wan, Shenglong Tang, Zhaogang Shu, Di Li, Shiyong Wang, Muhammad Imran, and Athanasios V Vasilakos. Software-defined industrial Internet of things in the context of Industry 4.0. IEEE Sensors Journal, 16(20):7373–7380, 2016.

[173] Jianguo Zhou, Hao Jiang, Jing Wu, Lihua Wu, Chunsheng Zhu, and Wenxiang Li. SDN-based application framework for wireless sensor and actor networks. IEEE Access, 4:1583–1594, 2016.

[174] Lucas MR Arbiza, Leandro M Bertholdo, Carlos Raniery P dos Santos, Lisandro Z Granville, and Liane MR Tarouco. Refactoring Internet of things middleware through software-defined network. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, pages 640–645. ACM, 2015.

[175] Clovis Anicet Ouedraogo, Samir Medjiah, Christophe Chassot, and Khalil Drira. Enhancing middleware-based IoT applications through runtime pluggable QoS management mechanisms. application to a oneM2M compliant IoT middleware. Procedia computer science, 130:619–627, 2018.

[176] José L Romero-Gázquez and M Bueno-Delgado. Software architecture solution based on SDN for an industrial IoT scenario. Wireless Communications and Mobile Computing, 2018, 2018.

[177] Faraz Idris Khan and Sufian Hameed. Software defined security service provisioning framework for Internet of things. arXiv preprint arXiv:1711.11133, 2017.

[178] Andrea Bianco, Robert Birke, Luca Giraudo, and Manuel Palacin. OpenFlow switching: Data plane performance. In 2010 IEEE International Conference on Communications, pages 1–5. IEEE, 2010.

[179] Alexander Shalimov, Dmitry Zuikov, Daria Zimarina, Vasily Pashkov, and Ruslan Smeliansky. Advanced study of SDN/OpenFlow controllers. In Proceedings of the 9th central & eastern european software engineering conference in russia, pages 1–6, 2013.

[180] Raphael Durner, Andreas Blenk, and Wolfgang Kellerer. Performance study of dynamic QoS management for OpenFlow-enabled SDN switches. In 2015 IEEE 23rd International Symposium on Quality of Service (IWQoS), pages 177–182. IEEE, 2015.

[181] Kiyohisa Ichino. OpenFlow communication system and OpenFlow communication method, December 10 2013. US Patent 8,605,734.

[182] David Erickson. The Beacon OpenFlow controller. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pages 13–18, 2013.

[183] Tie Luo, Hwee-Pink Tan, and Tony QS Quek. Sensor OpenFlow: Enabling software-defined wireless sensor networks. IEEE Communications letters, 16(11):1896–1899, 2012.

[184] Theo Kanter, Rahim Rahmani, and Arif Mahmud. Conceptual framework for Internet of thing' virtualization via OpenFlow in context-aware networks. International Journal of Computer Science Issues (IJCSI), 10(6):16, 2013.

[185] Mauro Conti, Pallavi Kaliyar, and Chhagan Lal. CENSOR: Cloud-enabled secure IoT architecture over SDN paradigm. Concurrency and Computation: Practice and Experience, 31(8):e4978, 2019.

[186] Martin Nagy. Software defined networking in wireless mobile networks. Information Sciences & Technologies: Bulletin of the ACM Slovakia, 11(1), 2019.

[187] Tie Luo, Shanhui Zhang, and Junjie Liu. Design of centralized control architecture for distribution network communication network based on SDN. In 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), pages 59–64. IEEE, 2019.

[188] Mahdi Nobakht, Craig Russell, Wen Hu, and Aruna Seneviratne. IoT-NetSec: Policy-based IoT network security using OpenFlow. In 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pages 955–960, 2019.

[189] Carlos Gonzalez, Salim Mahamat Charfadine, Olivier Flauzac, and Florent Nolot. SDN-based security framework for the IoT in distributed grid. In 2016 international multidisciplinary conference on computer and energy science (SpliTech), pages 1–5. IEEE, 2016.

[190] Nagarathna Ravi and S Mercy Shalinie. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. IEEE Internet of Things Journal, 7(4):3559–3570, 2020.

[191] Jianjun Sun, Jiaqi Yan, and Kem ZK Zhang. Blockchain-based sharing services: What blockchain technology can contribute to smart cities. Financial Innovation, 2(1):1–9, 2016.

[192] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress), pages 557–564. IEEE, 2017.

[193] Ashutosh Dhar Dwivedi, Gautam Srivastava, Shalini Dhar, and Rajani Singh. A decentralized privacy-preserving healthcare blockchain for IoT. Sensors, 19(2):326, 2019.

[194] Tivani P Mashamba-Thompson and Ellen Debra Crayton. Blockchain and artificial intelligence technology for novel coronavirus disease-19 self-testing, 2020.

[195] Mahtab Kouhizadeh, Sara Saberi, and Joseph Sarkis. Blockchain technology and the sustainable supply chain: Theoretically exploring adoption barriers. International Journal of Production Economics, 231:107831, 2021.

[196] Jun Wu, Mianxiong Dong, Kaoru Ota, Jianhua Li, and Wu Yang. Application-aware consensus management for software-defined intelligent blockchain in IoT. IEEE Network, 34(1):69–75, 2020.

[197] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. Blockchain for Internet of things: A survey. IEEE Internet of Things Journal, 6(5):8076–8094, 2019.

[198] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. Journal of Network and Computer Applications, 149:102481, 2020.

[199] Junfeng Xie, Helen Tang, Tao Huang, F Richard Yu, Renchao Xie, Jiang Liu, and Yunjie Liu. A survey of blockchain technology applied to smart cities: Research issues and challenges. IEEE Communications Surveys & Tutorials, 21(3):2794–2830, 2019.

[200] Bo Tang, Hongjuan Kang, Jingwen Fan, Qi Li, and Ravi Sandhu. IoT passport: A blockchain-based trust framework for collaborative Internet-of-things. In Proceedings of the 24th ACM Symposium on Access Control Models and Technologies, pages 83–92. ACM, 2019.

[201] Shreshth Tuli, Redowan Mahmud, Shikhar Tuli, and Rajkumar Buyya. FogBus: A blockchain-based lightweight framework for edge and fog computing. Journal of Systems and Software, 2019.

[202] Mathieu Boussard, Serge Papillon, Pierre Peloso, Matteo Signorini, and Erez Waisbard. STewARD: SDN and blockchain-based trust evaluation for automated risk management on IoT devices. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 841–846. IEEE, 2019.

[203] Mehran Pourvahab and Gholamhossein Ekbatanifard. An efficient forensics architecture in software-defined networking-IoT using blockchain technology. IEEE Access, 7:99573–99588, 2019.

[204] Abbas Yazdinejad, Reza M Parizi, Ali Dehghantanha, Qi Zhang, and Kim-Kwang Raymond Choo. An energy-efficient SDN controller architecture for IoT networks with blockchain-based security. IEEE Transactions on Services Computing, 2020.

[205] Pradip Kumar Sharma, Mu-Yen Chen, and Jong Hyuk Park. A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access, 6:115–124, 2017.

[206] Anichur Rahman, Md Jahidul Islam, Ziaur Rahman, Md Mahfuz Reza, Adnan Anwar, MA Parvez Mahmud, Mostofa Kamal Nasir, and Rafidah Md Noor. DistB-Condo: Distributed blockchain-based IoT-SD model for smart condominium. IEEE Access, 8:209594–209609, 2020.

[207] Bommareddy Lokesh and Narendran Rajagopalan. A blockchain-based security model for SDNs. In 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pages 1–6. IEEE, 2020.

[208] Qaisar Shafi and Abdul Basit. DDoS botnet prevention using blockchain in software defined Internet of things. In 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pages 624–628. IEEE, 2019.

[209] Kriti Bhushan and Brij B Gupta. Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. Journal of Ambient Intelligence and Humanized Computing, 10:1985–1997, 2019.

[210] Maninder Pal Singh and Abhinav Bhandari. New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges. Computer Communications, 2020.

[211] Saralees Nadarajah and Jeffrey Chu. On the inefficiency of bitcoin. Economics Letters, 150:6–9, 2017.

[212] Gavin Wood. A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151:1–32, 2014.

[213] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress), pages 557–564. Ieee, 2017.

[214] Schahram Dustdar, Stefan Nastić, and Ognjen Šćekić. Smart Cities – The Internet of Things, People and Systems. Springer, 2017.

[215] Raja Majid Ali Ujjan, Zeeshan Pervez, Keshav Dahal, Ali Kashif Bashir, Rao Mumtaz, and J González. Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Generation Computer Systems, 111:763–779, 2020.

[216] Jagdeep Singh and Sunny Behal. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. Computer Science Review, 37:100279, 2020.

[217] T Ninikrishna, Sutapa Sarkar, Richa Tengshe, Mahesh K Jha, Laxmi Sharma, VK Daliya, and Sudhir K Routray. Software defined IoT: Issues and challenges. In 2017 International Conference on Computing Methodologies and Communication (ICCMC), pages 723–726. IEEE, 2017.

[218] Eliseu Torres, Rafael Reale, Leobino Sampaio, and Joberto Martins. A SDN/OpenFlow framework for dynamic resource allocation based on bandwidth allocation model. IEEE Latin America Transactions, 18(05):853–860, 2020.

[219] Jianli Pan, Jianyu Wang, Austin Hester, Ismail Alqerm, Yuanni Liu, and Ying Zhao. EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts. IEEE Internet of Things Journal, 6(3):4719–4732, 2018.

[220] Sufian Hameed, Syed Attique Shah, Qazi Sarmad Saeed, Shahbaz Siddiqui, Ihsan Ali, Anton Vedeshin, and Dirk Draheim. A scalable key and trust management solution for IoT sensors using SDN and blockchain technology. IEEE Sensors Journal, 21(6):8716–8733, 2021.

[221] Ihsan H Abdulqadder and Shijie Zhou. Sliceblock: Context-aware authentication handover and secure network slicing using dag-blockchain in edge-assisted sdn/nfv-6g environment. IEEE Internet of Things Journal, 2022.

[222] Thiago Pereira da Silva, Thais Batista, Frederico Lopes, Aluizio Rocha Neto, Flávia C Delicato, Paulo F Pires, and Atslands R da Rocha. Fog computing platforms for smart city applications-a survey. ACM Transactions on Internet Technology (TOIT), 2022.

[223] Akshat Gaurav, Brij B Gupta, and Prabin Kumar Panigrahi. A comprehensive survey on machine learning approaches for malware detection in iot-based enterprise information system. Enterprise Information Systems, pages 1–25, 2022.

[224] Muhammad Aneeq Abid, Naokhaiz Afaqui, Muazzam A Khan, Muhammad Waseem Akhtar, Asad Waqar Malik, Arslan Munir, Jawad Ahmad, and Balawal Shabir. Evolution towards smart and software-defined internet of things. AI, 3(1):100–123, 2022.

[225] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. IEEE network, 32(1):96–101, 2018.

[226] Rishikesh Sahay, Weizhi Meng, DA Sepulveda Estay, Christian D Jensen, and Michael Bruhn Barfod. Cybership-iot: A dynamic and adaptive sdn-based security policy enforcement framework for ships. Future Generation Computer Systems, 100:736–750, 2019.

[227] Stefan Nastic, Hong-Linh Truong, and Schahram Dustdar. SDG-pro: a programming framework for software-defined IoT cloud gateways. Journal of Internet Services and Applications, 6(1):1–17, 2015.

[228] Bo Yi, Xingwei Wang, Keqin Li, Min Huang, et al. A comprehensive survey of network function virtualization. Computer Networks, 133:212–262, 2018.

[229] Mahdi Abbasi, Shima Maleki, Gwanggil Jeon, Mohammad R Khosravi, and Hatam Abdoli. An intelligent method for reducing the overhead of analysing big data flows in openflow switch. IET Communications, 2022.

[230] Esraa Mohamed. The relation of artificial intelligence with internet of things: A survey. Journal of Cybersecurity and Information Management, 1(1):30–24, 2020.

SHAHBAZ SIDDIQUI received MS Degree in Telecommunication from Hamdard University Karachi, Pakistan. He is pursuing his Ph.D. in Computer Sciences from the National University of Computer and Emerging Sciences, Karachi. He currently works as an assistant professor at the Department of Computer Science at the National University of Computer and Emerging Sciences in Karachi Pakistan. His research interests include the Internet of Things, SDN, and blockchain.

SUFIAN HAMEED received the Ph.D. degree in networks and information security from University of Göttingen, Germany. He works as assistant professor at Department of Computer Science at National University of Computer and Emerging Sciences, Pakistan. He also leads the IT Security Labs at NUCES. The research lab studies and teaches security problems and solutions for different types of information and communication paradigms. His research area includes network security, web security, mobile security and secure architectures and protocols for Cloud and IoTs.

SYED ATTIQUE SHAH received the Ph.D. degree from the Institute of Informatics, Istanbul Technical University, Istanbul, Turkey. During his Ph.D., he studied as a Visiting Scholar at the University of Tokyo, Japan, the National Chiao Tung University, Taiwan, and the Tallinn University of Technology, Estonia, where he completed the major content of his thesis. He has worked as an Associate Professor and the Chairperson at the Department of Computer Science, BUITEMS, Quetta, Pakistan. He was also engaged as a Lecturer at the Data Systems Group, Institute of Computer Science, University of Tartu, Estonia. Currently, he is working as a Lecturer in Smart Computer Systems, at the School of Computing and Digital Technology, Birmingham City University, United Kingdom. His research interests include big data analytics, the Internet of Things, network security, and information management.

IJAZ AHMAD received his MSc. and PhD. degrees in Wireless Communications in 2012 and 2018, respectively, from the University of Oulu, Finland. Currently, he is with the VTT Technical Research Centre of Finland. Dr. Ijaz has received several awards for his research work including the Nokia foundation, Tauno Tönning and the Jorma Ollila grant awards, and two IEEE best paper awards. He has been a visiting scientist at Aalto University, Finland in 2018, and at the TU Vienna, Austria, in 2019. His research interests include security in 5G and 6G, SDN and its security, and the applications of machine learning in future networks.

ADEL ANEIBA received his Ph.D. degree in the field of mobile computing and distributed systems from Staffordshire University, in 2008. He has worked as a Senior ICT Consultant for international organizations, for 10 years, including UNESCO and several governmental organizations for many years, and has participated in managing mega ICT projects mainly on data centre designing and development, and reengineering business processes. He is currently an Associate Professor in the area of computer networks and the Internet of Things (IoT) with Birmingham City University. He is the research lead for Cyber-physical Systems (CPS) Research Group. He is supervising several Ph.D. students on various research topics, such as the IoT, SDN, resources allocations, and optimization in 5G and Blockchain applications in the smart cities domain. His current research interests include the IoT, computer networks, evaluation, and optimization, and blockchain. He is a member of the Association for Computing Machinery (ACM) and the Institute of Engineering and Technology (IET) and is a Fellow of the Higher Education Academy (HEA) and a member of many technical committees for scientific academic conferences and journals.

DIRK DRAHEIM received a Ph.D. from Freie Universität Berlin and a habilitation from Universität Mannheim, Germany. From 2006-2008, he was area manager for database systems at the Software Competence Center Hagenberg, Austria. From 2008-2016 he was head of the data center of the University of Innsbruck and, in parallel, Adjunct Reader at the Faculty of Information Systems of the University of Mannheim. Currently, he is Full Professor of Information Systems at Tallinn University of Technology (Taltech), Estonia, and heading the Taltech Information Systems Group. The Taltech Information Systems Group conducts research in large- and ultra-large-scale IT systems, in particular, next generation of digital government technologies and digital government ecosystems. Dirk is co-author of the Springer book "Form-Oriented Analysis" and author of the Springer books "Business Process Technology", "Semantics of the Probabilistic Typed Lambda Calculus" and "Generalized Jeffrey Conditionalization". He is also an initiator and a leader of numerous digital transformation initiatives.

SCHAHRAM DUSTDAR is Full Professor of Computer Science heading the Research Division of Distributed Systems at the TU Wien, Austria. He has an H-index of 78 with some 36,000 citations. He holds several honorary positions: University of California (USC) Los Angeles; Monash University in Melbourne, Shanghai University, Macquarie University in Sydney, University Pompeu Fabra, Barcelona, Spain. From Dec 2016 until Jan 2017, he was a Visiting Professor at the University of Sevilla, Spain and from January until June 2017 he was a Visiting Professor at UC Berkeley, USA. He is founding co-Editor-in-Chief of ACM Transactions on Internet of Things (ACM TIoT) as well as Editor-in-Chief of Computing (Springer). He is an Associate Editor of IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, ACM Computing Surveys, ACM Transactions on the Web, and ACM Transactions on Internet Technology, as well as on the editorial board of IEEE Internet Computing and IEEE Computer. Dustdar is recipient of multiple awards: IEEE TCSVC Outstanding Leadership Award (2018), IEEE TCSC Award for Excellence in Scalable Computing (2019), ACM Distinguished Scientist (2009), ACM Distinguished Speaker (2021), IBM Faculty Award (2012). He is an elected member of the Academia Europaea: The Academy of Europe, where he is chairman of the Informatics Section, as well as an IEEE Fellow (2016) and an Asia-Pacific Artificial Intelligence Association (AAIA) Fellow (2021).