IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication October 00, 2020, date of current version October 00, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

SmartBlock-SDN: An Optimized Blockchain-SDN Framework for Resource Management in IoT

ANICHUR RAHMAN[1,2], MD. JAHIDUL ISLAM[3], ANTONIO MONTIERI[4] (Graduate Student Member, IEEE), MOSTOFA KAMAL NASIR[1], MD. MAHFUZ REZA[1], SHAHAB S. BAND[5,6], ANTONIO PESCAPÈ[4] (Senior Member, IEEE), MAHEDI HASAN[2], MEHDI SOOKHAK[7] (Senior Member, IEEE), AND AMIR MOSAVI[8,9,10]

[1]Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh (e-mail: anis.mbstu.cse@gmail.com; kamal@mbstu.ac.bd; shuvo_06_cse@yahoo.com)
[2]Department of Computer Science and Engineering, National Institute of Textile Engineering and Research (NITER), Savar, Dhaka, Bangladesh (e-mail: mahedihasan.ict@gmail.com)
[3]Department of Computer Science and Engineering, Green University, Dhaka, Bangladesh (e-mail: jahidul.jnucse@gmail.com)
[4]Department of Electrical Engineering and Information Technology, University of Napoli Federico II, Naples, Italy (e-mail: antonio.montieri@unina.it; pescape@unina.it)
[5]Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
[6]Future Technology Research Center, College of Future, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan (e-mail: shamshirbandshahaboddin@duytan.edu.vn)
[7]School of Information Technology, Illinois State University, Normal, IL, USA (e-mail: m.sookhak@ieee.org)
[8]Faculty of Civil Engineering, Technische Universität Dresden, 01069 Dresden, Germany (e-mail: amir.mosavi@mailbox.tu-dresden.de)
[9]John von Neumann Faculty of Informatics, Obuda University, 1034 Budapest, Hungary
[10]School of Economics and Business, Norwegian University of Life Sciences, 1430 Ås, Norway

Corresponding author: Amir Mosavi (e-mail: amir.mosavi@mailbox.tu-dresden.de); Shahab S. Band (e-mail: shamshirbandshahaboddin@duytan.edu.vn); Antonio Pescapè (e-mail: pescape@unina.it).

“This work in part is supported by the Alexander von Humboldt Foundation and the open access funding by the publication fund of the TU Dresden”

ABSTRACT Software-Defined Networking (SDN) and Blockchain are leading technologies used worldwide to establish safe network communication as well as build secure network infrastructures. They provide a robust and reliable platform to address threats and face challenges such as security, privacy, flexibility, scalability, and confidentiality. Driven by these assumptions, this paper presents an optimized energy-efficient and secure Blockchain-based software-defined IoT framework for smart networks. Indeed, SDN and Blockchain technologies have proven to be able to suitably manage resource utilization and to develop secure network communication across the IoT ecosystem. However, there is a lack of research works that present a comprehensive definition of such a framework that can meet the requirements of the IoT ecosystem (i.e. efficient energy utilization and reduced end-to-end delay). Therefore, in this research, we present a layered hierarchical architecture for the deployment of a distributed yet efficient Blockchain-enabled SDN-IoT framework that ensures efficient cluster-head selection and secure network communication via the identification and isolation of rouge switches. Besides, the Blockchain-enabled flow-rules record keeps track of the rules enforced in the switches and maintains the consistency within the controller cluster. Finally, we assess the performance of the proposed framework in a simulation environment and show that it can achieve optimized energy-utilization, end-to-end delay, and throughput compared to considered baselines, thus being able to achieve efficiency and security in the smart network.

INDEX TERMS IoT, SDN, Blockchain, Cluster Head Selection, Smart Technology, Flow-Rule Management, Network Security, Privacy.

I. INTRODUCTION

SOFTWARE-Defined Networking (SDN) is a prominent technology in the field of network communication that sums up a new dimension in the behavior of today’s network [1], [2]. A key advantage of SDN is that it allows programmability and flexibility to the emerging Internet of

Things (IoT) networks without any alteration of the architecture implementation [3]. However, on the other hand, it logically centralizes the network using the OpenFlow protocol. Also, as the number of IoT devices is growing speedily, the management and control of these devices need new concepts and mechanisms [4].

As distributed networks [5] are gaining more attention for managing security and are more appropriate than a centralized setup for the IoT ecosystem, *Blockchain* constitutes one of the most advanced and established technologies to secure online communications, to the point that it is usually referred as the Distributed Ledger Technology (DLT). In addition, Blockchain [6]–[8] provides a structure that records the digital information and distributes the data over the network but never allows for editing the information by a third party. As a consequence, many financial and data management systems effectively take advantage of the utility of Blockchain [9]. Cloud computing is also widely employed given that the vast majority of the systems requires a shared database where data can be simultaneously stored and retrieved through the Internet, on demand and with high availability [10] [11]. Indeed, the integration of cloud computing with IoT devices provides the latter both a greater and scalable (via the pay-as-you-go paradigm) storage capability and the connectivity that is needed to share information between the devices and make meaning from it at a fast pace [12]. Additionally, the IoT paradigm interconnects several physical appliances through the Internet and their number keeps rising day by day. The latest Cisco Internet Report [13] forecasts that Machine-to-Machine (M2M) connections will represent half of the global connected devices and connections by 2023. Specifically, it is estimated that 48% of M2M communications will be ascribed to connected home devices, while connected car applications will have a compound annual grow rate of 30% over the considered forecast period 2018–2023. Managing these huge amounts of devices (and the network providing them connectivity) is then becoming increasingly challenging. On top of this, IoT has vast security issues, exacerbated by the fact that the IoT environment is much more complex and heterogeneous than a traditional information technology infrastructure [14], [15].

In the earlier development stage of SDN-enabled IoT platforms, a centralized controller was used to manage the whole network, while addressing different IoT-ecosystem optimizations. Conversely, the usage of multiple controllers has been explored more recently [16], with the main goal of minimizing the packet loss. Also, the combination of Blockchain and SDN in IoT applications allows to heighten both privacy and security management [17], [18]. In addition, for guaranteeing communication reliability, the devices should be appropriately managed so as to deal with various issues as device failures. With this aim, mechanisms that allow the manufacturers to control the IoT data through the network have been proposed in literature [19]. More recently, *clustering methods* that intelligently manage the time slots have been presented. They allow to optimally and efficiently use sensors or IoT devices. When applying these cluster-based optimizations, many Blockchain-related factors that needed to be managed arise (e.g., software flow, privacy leakage, etc.). Moreover, the cluster head selection process is still a headache for researchers. Consequently, different works [21]–[25], [41]–[43] have presented several architectures (and observed the related issues) to manage and take advantage of the SDN-based IoT system obtaining heterogeneous outcomes.

Based on these reasons, the management of the resources of an IoT platform is indispensable to handle the major issues of SDN and IoT networks like privacy and security [20]. However, there are still several points to address for enabling Blockchain in an SDN environment employed to develop an IoT ecosystem. Therefore, in this paper, we seek answers to the following questions:

Q1. Distributed SDN-IoT Ecosystem: How can SDN-enabled IoT platform be efficiently deployed in a distributed network?

Q2. Cluster Head Selection: How can cluster head selection be optimized in the edge layer in terms of energy efficiency?

Q3. Distributed SDN Platform Security: What are the necessary security enhancements in an SDN-enabled distributed system?

With this aim, in the present study, we propose a layered hierarchical architecture for efficiently handling different issues of resource management in a distributed Blockchain-enabled SDN-IoT framework. We also develop a modern cluster head selection algorithm in the IoT layer that is faster and consumes lower energy with respect to considered baseline. Moreover, we leverage a Blockchain-enabled flow rules record that guarantees the consistency of the distributed controller cluster.

Paper Contributions and Organization

In view of the above-mentioned considerations, the contributions of this paper can be summarized as follows:

- We present a *layered hierarchy* to deploy a distributed yet efficient *Blockchain-enabled SDN-IoT framework*.
- We develop a *novel cluster head selection algorithm* in the IoT layer that, compared with the IEEE 802.15.4 baseline [58][1]
  1) has a *faster cluster head selection* procedure that combines both sorting and swapping techniques,
  2) attains *lower energy consumption* by comparing different conditions for taking the energy values.
- We develop a *Blockchain-enabled flow rules record* that keeps track of enforced rules and maintains consistency within the distributed controller cluster.

[1] This foundational RFC provides useful guidelines for the definition of IP-enabled Low-Power Wireless Personal Area Networks (WPAN), describing also the role and functions of the cluster-head in a generic WPAN. Conversely, more recent articles proposing cluster-head selection (cf. Sec. II) are usually tailored to specific scenarios/goals and/or do not provide critical details.

- We compare the *proposed Blockchain-enabled SDN-IoT architecture* with a classical Blockchain based on hashing and Proof of Work (PoW), showing that our proposal
  1) has *higher average throughput* and *optimized energy utilization*,
  2) has a *suitable transmission time* with respect to operations performed on the Blockchain.

The rest of the manuscript is organized as follows. Section II presents related works aligned with our paper. Then, Section III shows the layered architecture of our Blockchain-enabled SDN-IoT framework. Section IV and Section V discuss the threat model considered in the SDN environment and the cluster-head selection procedure employed in the IoT environment, respectively. Section VI covers the Blockchain-enabled SDN platform with flow rules verification. The implementation and experimental results are provided in Section VII and Section VIII, respectively. Finally, Section IX discusses lessons learned along with limitations and open challenges, followed by the conclusions provided in Section X.

II. RELATED WORK

Recently, several researchers have provided different contributions in fields related to emerging leading technologies such as smart networks, IoT, SDN, and Blockchain. This section presents a literature review of these works, along with the past studies that employed cluster head selection techniques in various scenarios.

A. SMART NETWORKS

Kazmi et al. [26] presented the concept of smart distribution network under the smart-grid paradigm and reviewed it from a planning viewpoint. Specifically, they highlighted the planning model of the smart distribution network, including implementation activities. In another research, Huang et al. [27] proposed an architecture for smart networks including intelligent mechanisms aided by big wireless data (BWD), artificial intelligence (AI) methods, as well as network function virtualization techniques. The authors also allowed mobile users to connect to the best network (with a manageable cost) achieving goal Quality-of-Service (QoS). On the other hand, Takenaka et al. [19] discussed how manufacturers can take advantage of IoT data showing also a smart network example. This paper stressed the usefulness of an appropriate data format and analytical methods for objectives like mass customization or creation of new services. A recent study by Chakrabarty et al. [28] devised an architecture for secure smart cities. In detail, they introduced four fundamental IoT architectural components, i.e. Trusted SDN controller, Black Network, Registry Unification, and Key Management System.

B. CLUSTER HEAD SELECTION

Kumar et al. [29] proposed a technique for clustering which divide a large network into small clusters where each one has its Cluster Head (CH). These CHs employ the Time Division Multiple Access (TDMA) method for supplying the time slots to every node. Similarly, in another research, Angel et al. [30] devised an Enhanced Energy Efficient Clustering Algorithm (EEECA) for reducing the energy consumption needed for picking the CHs in Mobile Wireless Sensor Networks. Simulation results showed that the EEECA performs better than the existing EECA-M2 algorithm [31] used as baseline. Al-Baz et al. [32] proposed a novel variant of the LEACH protocol called Node Ranked–LEACH, which is heightened to enhance the network's lifespan by relying on the algorithm of node rank (NR). The authors have proposed to solve the random process selection as an algorithm, which in other LEACH versions, leads to unintended failure for specific CHs. An analogous study is accomplished in the work by Zhao et al. [33] that introduced an amended LEACH-based cluster-head selection algorithm for Wireless Sensor Networks (WSNs). In simulation phases, the authors took into account different networking aspects like network lifetime, energy conservation, and the amount of data transferred.

C. SDN-BASED WSN WITH CLUSTERING

Murugaanandam et al. [34] leveraged a CH-selection procedure to prolong the lifetime of WSNs. They proposed the RE-TOPSIS protocol, that combines the conventional LEACH protocol with fuzzy logic using a multi-criteria decision making approach for CH selection. Specifically, six different criteria (i.e. residual energy, neighbors' availability, energy utilization rate, node and base station distance, and node reliability) are considered to reliably select the CHs. Simulations showed that the proposed scheme effectively enhances the network lifetime reducing the frequency of CH selection by $\approx 20\%$ as compared to the conventional LEACH protocol. Shafique et al. [35] proposed SADFIR, an interactive clustering routing protocol that enables distributed SDN controllers to collaborate with forwarding network devices for routing the sensed information. SADFIR iteratively manages the reconfiguration of network settings based on the traffic of IoT devices deployed in the infrastructure layer of the proposed architecture. Self-reconfiguration of a node depends on environmental temperature, humidity, and pressure that are compared with threshold values chosen by the network administrator. The authors asserted that the proposed SADFIR outperforms state-of-the-art routing protocols.

Recent researches have proposed energy-efficient selection of CHs in WSNs that are picked out comparing the energy level of the nodes. In [36], the authors devised an efficient CH-selection scheme that rotates the CH positions over the time among the IoT devices with higher residual energy, whereas in [37] an hybrid optimization algorithm with multi-objective constraints (involving distance, energy, and delay) is presented. Both approaches showed good performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3058244, IEEE Access

in terms of energy utilization and network lifetime. More recently, Ouhab et al. [38] designed a modeling paradigm based on multi-hop clustering technique used to organize the sensors in clusters with the aim of reducing the energy consumption in large-scale SDN-based IoT networks. Model simulations exhibited better end-to-end delay, packet delivery ratio, and energy consumption than conventional solutions.

WSNs need efficient data aggregation since sensors frequently capture data that can contain a significant amount of noise and redundant information. To mitigate this problem Ullah et al. [39] proposed a data-aggregation scheme based on node clustering that leverages data similarity and density. Instead of contributing to the CH-selection, they applied filtering procedures to reduce the noise in data before sending them to the CH and extreme learning machine to aggregate data in the CH. Simulation results showed that the proposed scheme attains good performance in terms of clustering accuracy, energy efficiency, and number of living nodes.

D. IOT WITH SDN

In [40], Matheu et al. addressed the Manufacturer's Use Description (MUD) model for network access control, data privacy, as well as channel and authorization protection policies. They then employed the SDN platform for efficiently accessing device data and resources, and also used the Blockchain technology to share data/information through Hyperledger[2] with the help of IoT devices. Moreover, Molina et al. [41] presented a security framework for the continuous and on-demand management of virtual Authentication, Authorization, and Accounting (AAA) in SDN-enabled IoT networks. The authors achieved scalable bootstrapping of IoT devices and fine-grain management of their access control to the network. Differently, Conti et al. [42] presented a novel combination of cloud computing, IoT, and SDN resulting in the devised CENSOR framework, which is leveraged to provide a secure floor in the IoT scenario. With this aim, CENSOR encompasses a reliable and secure IoT-network architecture enabled by the cloud and based on the SDN technology. The work highlighted also a number of challenges and possible threats that should be addressed, such as advanced security—e.g., against Distributed Denial of Service (DDoS) attacks—suitable routing algorithms, and proper network scalability. Abdelaziz et al. [43] recommended a distributed controller cluster to handle reliability, scalability, fault tolerance, and interoperability issues in an SDN. The authors also claim that their proposed method achieves reasonable CPU utilization and thus optimizes the controller performance. With a specific focus on IoT-application security, Liu et al. [44] designed Middlebox-Guard (M-G), an SDN-based data transfer security model for dealing with various attacks and improving the stability of the network. First, the authors address the placement of middleboxes (related to a set of defined security policies) via a placement selection algorithm, then two SDN-resource control algorithms are leveraged to satisfy the coverage requirements under switch volume constraints. The simulation results showed that the devised M-G model could improve safety and stability of the IoT network.

E. BLOCKCHAIN FOR SDN

Yazdinejad et al. [46] presented an IoT architecture that efficiently combines SDN and Blockchain leading technologies. Their aim is to apply this architecture to the SDN controllers of IoT networks leveraging a cluster structure with a novel routing protocol to mitigate networking challenges such as security, privacy, confidentiality, and so on. In detail, the authors mainly focused on the energy-efficient mechanisms for file transferring between the IoT devices in an SDN platform. The architecture employed both public and private Blockchains for (peer-to-peer) communication between the IoT devices and SDN controllers, together with a distributed-trust authentication method. Similarly, Chaudhary et al. [47] leveraged the Blockchain and SDN for increasing the QoS of the network in an intelligent transportation system. Specifically, they devised BEST, a Blockchain-based secure energy trading scheme for electric vehicles. BEST used Blockchain to validate vehicles' requests in a distributed fashion, hence avoiding the single point of failure. Simulation results showed that the SDN architecture successfully integrated the Blockchain by enhancing the network QoS, while energy utilization is also more efficient in the devised deployment. Nevertheless, the authors did not take into account different energy sources. El Houda et al. [48] presented a Blockchain-based architecture, named Cochain-SC, that allows multiple SDN-based domains to securely collaborate and transfer attack information in a decentralized manner, and combines intra-domain and inter-domain DDoS mitigation. The authors calculated the performance of Cochain-SC in terms of efficiency, flexibility, security, cost effectiveness, and detection accuracy of illegitimate flows. Ferrag et al. [49] presented an overview of applications of the Blockchain technologies in various IoT areas, e.g., Internet of Vehicles, Internet of Energy, virtual web, cloud and edge computing, and so on. In this survey, the authors discussed also the five most common attacks in IoT networks, namely identity-based, cryptanalytic, reputation-based, manipulation-based, and service-based attacks. They also defined a taxonomy of the state-of-the-art methods for attaining secure and privacy-preserving Blockchain technologies and compared them on the basis of the specific model, security goals, performance, computation complexity, limitations, and communication overhead. In [54], Sharma et al. designed the "DistBlockNet" framework towards secure distributed SDN architecture for IoT via the Blockchain technology. They brought out a scheme for updating and validating the flow rule table using Blockchain. The experimental evaluation demonstrated the effectiveness of DistBlockNet in terms of accuracy, scalability, defense effects, and performance overhead incurred.

In summary, a number of works have proposed different solutions so far to enable Blockchain in an SDN-based IoT

[2] https://www.hyperledger.org/

ecosystem. However, there is still a lack of comprehensive scenarios provided by any of these works. Therefore, in the following, we devise and discuss a comprehensive and optimized framework that ensures security in a software-defined IoT ecosystem.

III. LAYERED ARCHITECTURE

In Fig. 1, we depict the Layered Architecture of the Blockchain-enabled SDN-IoT ecosystem that aims to optimize the Blockchain-based SDN framework for enhancing resource management in IoT network. The Layered Architecture is organized in three distinct layers, that is Perception Layer, Edge Layer, and Cloud Layer, related to just as many environments, namely IoT Environment, SDN Environment and Blockchain Environment.

At the beginning, the *IoT Environment* (viz. the *Perception Layer*) contains the IoT sensors and devices which are responsible for sensing the data in real time and transmit them into the next sublayer (also part of the Perception Layer). The latter aims to select the CHs with higher energy. Contextually, IoT forwarding devices (e.g., switches, routers, phones, storage devices, etc.) provide the sensors' data to the CHs selected among the arbitrary clusters. This process is controlled by Access Points (APs) that finally forward all sensors' information to the SDN Environment.

In the actual *SDN Environment*, the *Edge Layer* is structured by two conventional levels, namely the data plane and control plane. IoT devices (e.g., routers, switches, firewalls, storage devices, etc.) can forward data through SDN common gateways (i.e. SDN-IoT gateways in Fig. 1). Hence, multiple SDN controllers manage and possibly filter the IoT devices' data dynamically: this action is executed via the OpenFlow protocol.

Finally, the *Cloud Layer* comprises with *Blockchain Environment* and data centers where data are transmitted through the cloud network. In detail, cloud computing provides the real-time shared database, while Blockchain helps to communicate with each of the data as a block by block in the networking system, along with providing extra security, privacy, and confidentiality among the data blocks (i.e. in the network). Furthermore, Blockchain realizes also the chain for communicating one block with other blocks suitably.

IV. THREAT HANDLING

To enhance the security of the SDN-IoT ecosystem, attacks generated from both the inside and outside of the SDN environment should be identified and defeated. For instance, network devices that have become compromised should be automatically detected and quarantined before they can negatively affect the network, availability services should be implemented to enhance the application performance against system and network failures, including through the integration of legacy security applications.

Figure 2 shows the threat model that we consider to deal with these attacks. Specifically, in our layered architecture we propose to leverage an Attack Mitigation System (AMS) that works combinedly on attacks from both the inside and outside of SDN. In the back-end of the AMS, various algorithms, which automatically detect and handle the attacks, are implemented. Then, the *AMS_1* module in the SDN gateway handles the attacks from the outside of SDN, while the *AMS_2* and *AMS_3* handle the attacks from the inside of SDN.

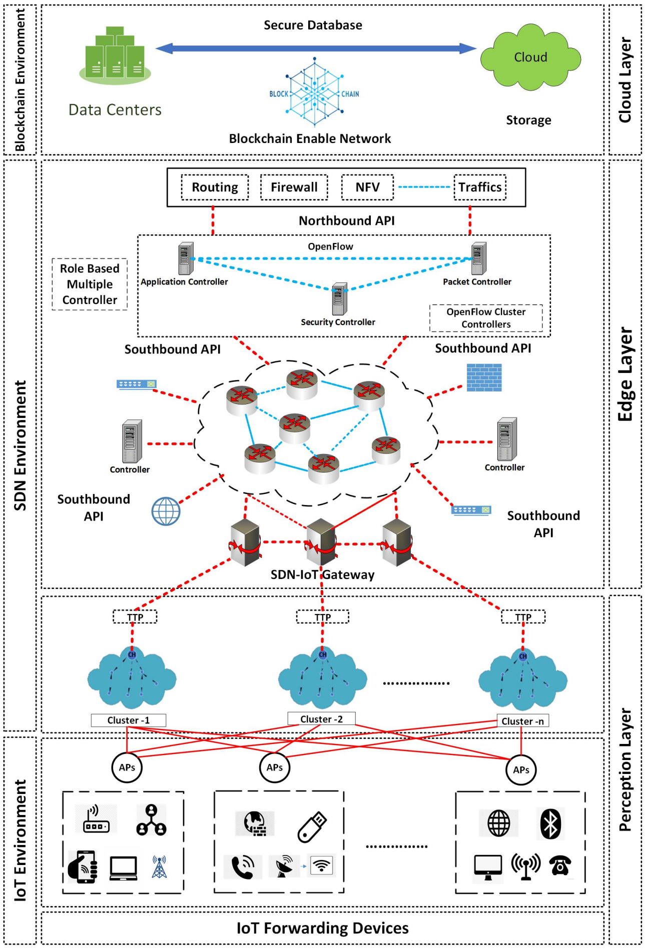


FIGURE 1. Layered architecture of Blockchain-enabled SDN-IoT ecosystem.

Indeed, there exist different types of attack that are aimed at reducing the reliability and availability of the proposed layered architecture. Notably, the most relevant attacks that should be handled are *flow table overloading* attacks, *topology poisoning* attacks, *saturation* attacks, and *Distributed Denial of Service* attacks [50]–[53]. Therefore, *we propose to integrate in our layered architecture different algorithms* presented in state-of-the-art works to address the SDN-security issues that are discussed hereinafter.

Flow Table Overloading Attack

The main culprit of the *flow table overloading* attack is to occupy space of the flow table of an SDN switch intentionally. As a result, the switch would not be able to provide the necessary network services. In [50], the authors created

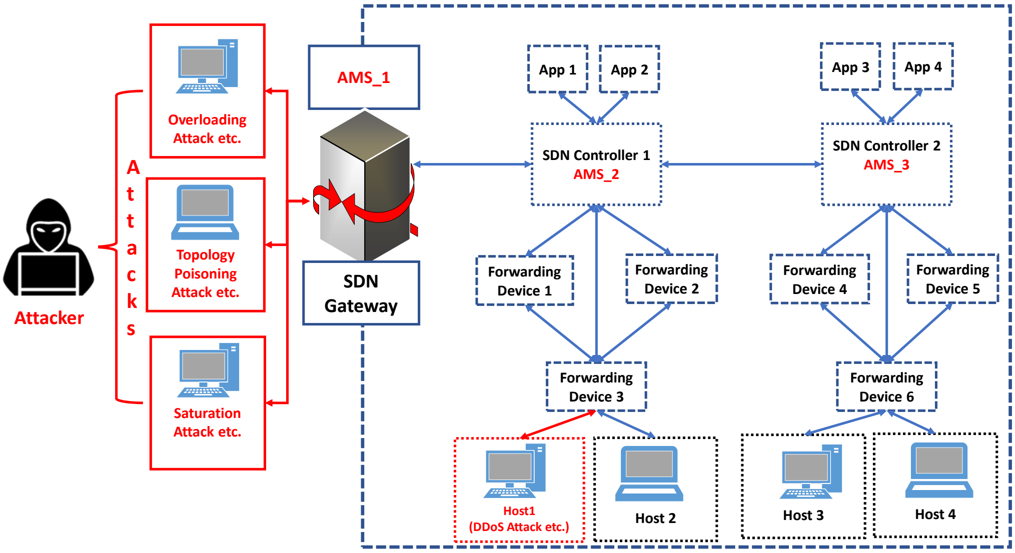


FIGURE 2. Threat model employed in the SDN environment of the proposed layered architecture. Different state-of-the-art methods could be integrated in the Attack Mitigation System (AMS) to detect and mitigate the attacks against the SDN.

SDN switches with 8k flow rules per second. To show the effect of the flow table overloading attack, they recorded the time needed to overpower the switches under different attack rates and found that the switches are overwhelmed within 10 seconds if the attack rate reaches 800 requests per second. They proposed a strategy named *peer support strategy* that integrates idle flow table resources in the SDN system to mitigate the flow table overloading attacks in the switches and minimize QoS degradation.

Topology Poisoning Attack

The *topology poisoning* attack has the malicious task of misleading the topology-discovery service of the control plane by spreading misinformation (viz. injecting fake links). As a result, it leads to a significant increment of the packet-loss rate. In [51], the authors devised an efficient approach named *TopoGuard* being a security extension of the OpenFlow controller that automatically detects topology poisoning attacks in real-time.

Control Channel Saturation Attack

The *control channel saturation* attack overloads the control plane intentionally (e.g., by triggering several table-miss packets), thus exhausting the controller's resources and delaying the forwarding of messages to the OpenFlow switches. In [52], the authors used a method named *LineSwitch* to deal with control plane saturation attacks. It deploys proxying and blacklisting of network traffic (directed to the control plane) based on the probability of saving the control plane from overwhelming (e.g., resiliency against SYN-flooding saturation and buffer saturation vulnerabilities).

Distributed Denial of Service Attack

A *Distributed Denial of Service (DDoS)* [45] is an attack in which multiple malicious attackers deliberately keep the network busy so that denying the service to legitimate users or systems. In [53], the authors employed different *Machine Learning (ML)* algorithms to detect DDoS attacks. Among the tested methods, they proved the Multi-Layer Perceptron performs the best with up to 95% detection rate. However, to identify and mitigate DDoS attacks also other ML models (e.g., Random Tree, Random Forest, Support Vector Machines, etc.) proved to be effective.

V. CLUSTER HEAD SELECTION PROCEDURE

As described above, Internet of Things devices (e.g., switches, routers, firewalls, intelligent storage devices, etc.) forward data by means of SDN-IoT enabled gateways; the SDN dynamic controller is capable of refining the IoT sensors' data; the OpenFlow protocol assists the latter process. However, the preceding steps can efficiently accomplish their distinct functions only if the IoT devices are able to set up a cluster head correctly.

A. PRELIMINARIES

Hundreds or even thousands of sensor hubs are connected in a wireless sensor network that performs in low power, while the sensors (e.g., IoT devices) are normally multi-functioning. Each sensor node consists of a data transceiver, micro-controller, and of course, an energy source, which is usually a battery. All these components work unitedly to form the network.

Sensor nodes have confined energy, whereas the base station has no energy limitations, but it remains far away from the sensor nodes. In SDN, the nodes are not movable and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3058244, IEEE Access

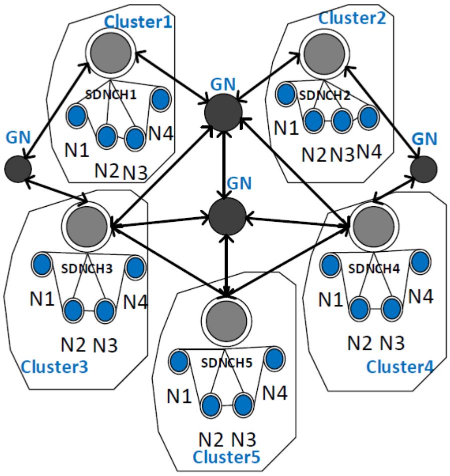


FIGURE 3. Cluster-head selection process in an SDN environment [18].

they constantly have data streams to send to the base station. Hence, the sensors' energy is exhausted by collecting data and sending them to the base station. One of the really crucial issue in the sensor-network area is then power consumption.

As wireless communication and data-transmission technology via sensors have become increasingly popular during the last decade, the headache of researchers to mitigate the uses of energy in real-time application has significantly grown. The best way to lessen the power consumption is to apply the clustering technique. Enforcing clustering, a set of sensors could be selected, which will be used as data transmitters to the base station. These sensor nodes are known as Cluster Heads (CHs), and only these heads communicate with the station. The other sensors send their information to the base station via the CH of their area. Firstly, a clustering algorithm is used to constitute the clusters of sensors, then a cluster-head selection algorithm is leveraged to choose the CH among the sensors into each cluster found by the clustering algorithm.

B. IMPORTANCE OF CLUSTER HEAD SELECTION

In this section, we discuss the importance of the cluster-head selection algorithm, whose fundamentals are depicted in Fig. 3. In this research, we combine both sorting and swapping techniques to steadfastly elect the CH. We also attain low energy consumption by comparing different conditions for taking the potential values.

Indeed, since the sensor in a cluster sends its information to the base station via the CH of that cluster, if there is a long distance between the CH and each sensor, the latter consumes high power to cover this long distance. Therefore, the CH should be selected so that to minimize the absolute distance between itself and the other members of a cluster.

We highlight that the CH-selection algorithm proposed hereinafter is independent from the specific clustering method used to group the sensors. Nevertheless, the chosen clustering method should measure the similarity among the sensors via a distance metric (e.g., Euclidean distance, Manhattan distance, Minkowski distance, etc.). Fuzzy C-means, K-means, DBSCAN are notable examples of algorithms that could be possibly employed.

Moreover, there exists some cluster-head selection algorithm already established like IEEE 802.15.4. But till now, it is challenging to pick out the right CH, which will utilize the energy efficiently. In this regard, Fig. 3 shows how a CH is selected to transmit data packets over the network. As mentioned before, each (IoT) device (viz. sensor) could carry out this transmission but it would consume a good amount of energy to send and receive the data to and from the base station. Consequently, arises the need for clustering sensor nodes and further for choosing among the nodes of a certain cluster one that acts as the CH. Indeed, the CH is the only is charge of communicating with the base station and of sending and receiving data to and from the other nodes belonging to the same cluster. Thus, selecting the CH so as to consume lesser energy for transmitting the data and be able to communicate with the other nodes into the cluster is very crucial for SDN-enabled IoT networks.

In summary, the CH should have the capability of transmitting data from the sensors (belonging to the same cluster) to the base station and vice versa. It should be also capable of enhancing the lifetime of the network (viz. reducing the energy consumption) efficiently. To this aim, the cluster head selection algorithm starts from the initial energy and optimizes the values of CHs to choose the next group of CHs for the network that suits for IoT forwarding devices such as smart cities, building, healthcare, and other related intelligent systems.

C. PROPOSED ALGORITHM FOR CLUSTER-HEAD SELECTION

Terminologies

For better understanding the algorithm for cluster-head selection proposed herein, Tab. 1 reports symbols and terminologies used in the following. These represent the parameters and factors employed to compute the desired CHs. It is worth noting that in the following, we will use node and sensor terms interchangeably.

Flowchart

For further clarity, we provide a simple flowchart in Fig. 4 depicting the operational steps of proposed algorithm. First, the target number of clusters is computed, as one CH will be

TABLE 1. Symbols used in the definition of cluster-head selection algorithm.

Symbol	Definition
n	Number of nodes in the list
N	List of nodes
$N[i].energy$	Energy of the i^{th} node of list N
LDS	Lowest Distance Separation
G_{dist}	Geographical Distance
$E_{station}$	Energy needed to communicate with the base station

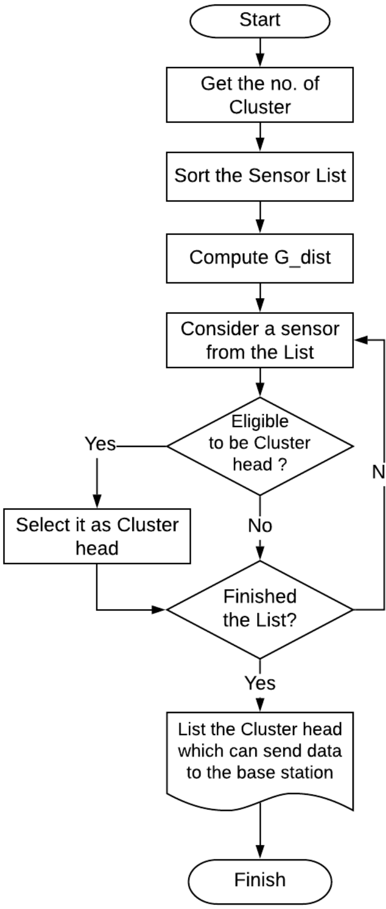

Start
Get the no. of Cluster
Sort the Sensor List
Compute G_dist
Consider a sensor from the List
Eligible to be Cluster head ?
Yes
No
No
Select it as Cluster head
Finished the List?
Yes
List the Cluster head which can send data to the base station
Finish

FIGURE 4. Flowchart of the proposed cluster-head selection algorithm.

selected from each cluster. Then, the sensor list is sorted to ease the processing of the nodes. G_{dist} indicates the gravity distance of the nodes into a cluster.

The CH is normally selected with the minimum distance and highest energy level. Hence, each of the sensors from the list is considered one by one and checked for eligibility to be the CH. If a sensor satisfies the conditions for being the head, it is picked out as the CH for that cluster, conversely the algorithm scans another sensor from the list. As soon as the sorted list of the sensors is fully scanned, the list of CHs is returned. The CHs are the only nodes that can perform data transmission with the base station.

Algorithm

By utilizing the algorithm depicted in Fig. 4, our principal objective is expediently choosing the head of each cluster among the general cluster nodes in order to effectively save sensors' energy. The pseudocode of the proposed cluster-head selection algorithm is given in Alg. 1.

Algorithm 1 Proposed Cluster-Head Selection Algorithm.

```
    Input: Total nodes ($n$), List of nodes ($N$)
    Output: List of Cluster Heads ($CH$), Outing Route ($OR$)
 1: while true do
 2:   for $i \leftarrow 1$ to $n - 1$ do
 3:     $min \leftarrow i$
 4:     for $j \leftarrow i + 1$ to $n$ do
 5:       if $(N[j].energy < N[min].energy)$ then
 6:         $min \leftarrow j$
 7:       end if
 8:     end for
 9:     $swap(N[i], N[min])$
10:   end for
11:   $SLNs \leftarrow N$                          ▷ Sorted list of nodes
12:   $LDS \leftarrow G_{dist}(SLNs)$     ▷ Computing Lowest Distance
    Separation ($LDS$) by means of Geographical Distance
13:   for $i \leftarrow 1$ to $n$ do
14:     if $(N[i].energy\ is\ Max\ \ \&\ \ LDS)$ then
15:       $CHs \leftarrow N[i]$                ▷ Selecting Cluster Heads
16:     end if
17:   end for
18:   $\beta \leftarrow E_{station}$     ▷ Minimum amount of energy a CH needs
    to communicate with the base station
19:   for $i \leftarrow 1$ to $len(CH)$ do
20:     $CH[i].send\_request()$ ▷ Sending the “request to send
    data”
21:     if $(CH[i].energy \geq \beta)$ then
22:       $CH[i].send\_data()$                    ▷ Sending data
23:     end if
24:   end for
25: end while
```

The steps describing the cluster-head selection algorithm are outlined in the following[3]:

S.1: The cluster nodes are estimated by dividing the whole number of nodes by the number of desired heads (viz. the number of clusters)[4]

S.2: The nodes are sorted according to their energy values by comparing the nodes' energy (lines 1–5) and swapping them if needed (lines 6–9).

S.3: The list of sensor nodes $SLNs$ sorted based on their energy values is obtained (line 11).

S.4: The Lowest Distance Separation (LDS, namely the minimum space between two sensors) is computed by considering the Geographical Distance G_{dist} of the nodes within $SLNs$. In this case, G_{dist} is calculated using the Euclidean distance[5].

S.5: The node with the highest energy is referred to as the CH, while the other nodes remain (simple) members of the cluster (lines 12–17). In this step, the energy values of the cluster nodes (and consequently the selection of CHs) are harmonized depending upon the G_{dist} (i.e. the CH selection does not depend only on the nodes' energy level but also on the LDS) with the aim of optimizing the energy consumption using the selected CH.

[3] Line numbers refer to Alg. 1.

[4] The number of clusters strongly depends on network topology and IoT devices' deployment and it is preliminarily decided by the network administrator.

[5] It is worth noticing that other distance metrics can be possibly employed (e.g., Manhattan distance, Minkowski distance, etc.).

S.6: The previous step is repeated to choose the CHs of all the clusters that are stored in an array-like structure.
S.7: The energy cost β (i.e. the minimum amount of energy a CH needs to communicate with the base station) is assigned as a real and positive constant so as to guarantee the success of the communication (line 18).
S.8: All the CHs send a "request to send data" to the base station (line 20) and wait for an acknowledgment (ACK).
S.9: If the CH receives a positive ACK, that means it has sufficient energy to send data (line 21).
S.10: Then, in the affirmative case, the CH can send data to the base station (line 22).

It is worth noting that only eligible CHs take the permission for routing data into a suitable path (OR). Besides, some additional information is reported to indicate the purpose of the operation performed.

VI. BLOCKCHAIN-ENABLED SDN PLATFORM

We propose two different Blockchains for the control layer and the data layer. Blockchain in the control layer contains the distributed flow rules and maintains the consistency of the flow rules of each cluster. In detail, the chain logs all the updates, thus resulting in a version control management system in the control layer. On the other hand, Blockchain in the data layer works differently. All the switches dump their flow rules in the chain sequentially and verify if they are maintaining the same rule set. If any of the switches do not dump the same rules, the record is not updated, and the switch is isolated from the environment. This isolation helps to identify not only a fault in the switch but also to contain adversaries if the switch is compromised.

In view of these considerations, we have divided the Blockchain workflow for the layered architecture (see Fig. 1) into two parts based on common SDN layers: *control layer* and *data layer*. Hereinafter, we present the workflows designed to ensure flow-based rouge node detection and security in the network.

A. BLOCKCHAIN-ENABLED CONTROL LAYER

The control layer enables the Blockchain-based distributed ledger to keep track of all the switches working correctly and maintaining the same flow rules. It is also responsible for version control for future history-checking capability. In the following, we describe these features in detail.

- **Distributed Flow Rules**: the (OpenFlow) controller cluster (see Fig. 1) maintains a distributed ledger to update the flow rules and to broadcast new rules towards all the controllers, thus guaranteeing the *consistency* of the controllers' rule set. The ledger is updated by the system administrator using a *REST API*.
- **Version Control Management System**: the distributed ledger works as a version control management system that keeps track of the updates regarding all the new flow rules. Immutable previous history can be accessed through *REST API* calls. Distributed flow rules are initially set up when the controllers are run for the first time. The initial rules are kept in the *genesis block* of the control-layer Blockchain. Later, further flow rules are updated in the chain, with each block representing a new version of the flow rule. New rules are added to a new block called the *up-to-date block*. The application layer maintains the update process using a *REST API*[6].

B. BLOCKCHAIN-ENABLED DATA LAYER

The data layer is primarily responsible for providing flow-rules dump and verification along with isolation of rouge switches. Hereinafter, we give details of the functionalities of the Blockchain-enabled SDN data layer.

- **Flow-rules dump**: we introduce a Blockchain that works in the data layer and periodically updates a block only if all the switches agree on all the rules. In detail, the switch flow-rule tables can be dumped in two ways. First, the controllers can be configured at the initial stage to broadcast messages towards the switches to dump flow rules in a file or towards an application. The second solution is to write an application that invokes Open vSwitch commands to collect the flow-rule tables and compare them to maintain the consistency. In the implementation of our framework, we consider the latter approach to reduce the controller-cluster load and the computational complexity in the controller nodes.
- **Flow-rules verification**: we have designed an application that periodically collects the dump from the switches and matches all dumped data against the updated version of the flow rules included in the controller Blockchain. We convert both entries to a hash and match one against the other. Considering hash has a significant advantage: it is both convenient and secure. Also, to maintain the accuracy and to avoid duplication issues of the same data, we store the hash of the whole dump in the chain.
  To verify the flow rules in an SDN switch, two state-of-the-art fault-detection algorithms have been proposed in [55]: *forwarding detection* and *weighting detection*. Forwarding detection involves the process of checking if the intended packets are being forwarded to the right destination. Differently, weighting detection considers the statistical weight of several packets to be close to their expected values. As depicted in Fig. 5, here we use the forwarding detection method by configuring a firewall or a (stealthy) Intrusion Detection System (IDS) [56] according to the flow rules determined by the controller and check if the switch is forwarding the packets accordingly.
- **Switch Isolation**: while matching the dump data from the switches, each switch is disconnected from its peers if the hash of the flow tables does not match the hash of the updated block in the control-layer chain. We consider the hash of the dumped data to make the matching

[6] All SDN controllers—including OpenDaylight, POX/NOX, Ryu, HP VAN, etc.—support REST-based API to communicate with the controller from the application layer.

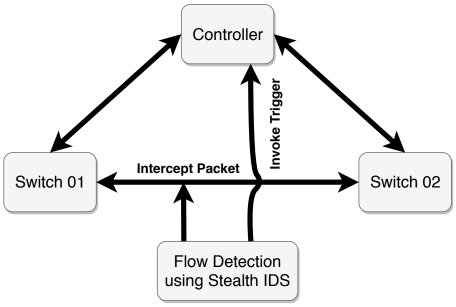


FIGURE 5. Stealthy IDS setup for detecting flow-rule violation.

process convenient. Additionally, the hash exhibits a significant change if even a single byte is changed in the flow information. The isolation is attained by invoking a new rule and sending it to the control-layer Blockchain. The new rule blocks the interfaces connected to the compromised switch. Then, the new rule is broadcast again to all the switches in the cluster that maintain the new rules.

VII. IMPLEMENTATION AND EXPERIMENTAL SETUP

This section presents the implementation choices regarding the integration of the Blockchain technology into the SDN control and data layers in Sec. VII-A. Then, in Sec. VII-B, we detail the experimental setup we have leveraged along with the parameters of the simulation environment.

A. IMPLEMENTATION OF THE BLOCKCHAIN-ENABLED SDN

We have implemented a *RESTful application* that collects switch flow entries using the *GET* request from an SDN controller. In this work, we use the *Ryu controller* written in python and with built-in REST API access. Also, we have realized a Blockchain-based ledger that is accessible through a REST API as well. To implement OpenFlow switches and network topologies, we leverage the Mininet-WiFi emulator, in which we code the topologies using python. In detail, we employ Postman, an API building and testing tool that uses HTTP requests (i.e. GET, POST, PUT, PATCH) to obtain and update OpenFlow switch rules in the SDN environment. Finally, we run our simulation with a setup configured according to our proposed model (see Fig. 1).

The Blockchain-based ledger we leverage is also equipped with a REST API for the communication with our application. In each block, we include an index, a timestamp (Unix timestamp[7]), newly installed flow-rule details, and the hash of the previous block. A simple JSON-based[8] flow-rule is encoded as shown by the code snippet in Listing 1.

Listing 1. Example of a JSON-encoded flow rule.

```
{
    "dpid": 1,
    "cookie": 1,
    "cookie_mask": 1,
    "table_id": 0,
    "idle_timeout": 30,
    "hard_timeout": 30,
    "priority": 11111,
    "flags": 1,
    "match":{
        "in_port":1
    },
    "instructions": [
        {
            "type": "APPLY_ACTIONS",
            "actions": [
                {
                    "max_len": 65535,
                    "port": 2,
                    "type": "OUTPUT"
                }
            ]
        }
    ]
}
```

We check the flow-rules from switches using the *GET* request. Using *PostMan*, a REST API testing tool, we collect the stats of switches with the request- “$GET\ http : //localhost : 8080/stats/desc/1$”.

1) SDN Environment Consistency

As we are maintaining the same rule set for all the switches in our network (viz. homogeneous rules), we also need to guarantee the *consistency*, namely that every switch is following identical rules. To ensure this we scan the flow rules by receiving their hashes from all the switches. For instance, we receive the following hash:

65d1c600e16d24e4a79d7e0cecb8f71a283b8e2775c8f88dffc8fb3768

Then, we compare the received hash with the existing hash stored in the corresponding block and check if they match.

2) SDN Environment Security

The *security* in our framework is achieved by isolating a rouge switch from the distributed network. To enable the isolation, we configure the controller to allow and broadcast new flow rules, and to disconnect the rouge switch from the network. While broadcasting the new rule, a hash is generated and stored once again in the Blockchain. If we need to integrate the switch within the network again, we inject new rules and store the hash in the ledger simultaneously.

[7] Number of seconds counted since the *Unix Epoch* (00:00:00 UTC, January 01, 1970).

[8] JSON stands for JavaScript Object Notation. JSON format is usually employed to encode the data exchanged with REST APIs.

TABLE 2. Parameters of the simulation environment grouped by reference technology.

Simulation Parameter	**Value**
General Parameters	
Network emulator	Mininet-WiFi
Cloud storage platform	OpenStack
Packet analyzer	Wireshark
Programming language	Python
SDN Parameters	
SDN routing protocol	OpenFlow
Number of SDN controllers	5
Blockchain Parameters	
Blockchain platform	Ethereum
Consensus protocol	Proof of Work (PoW), Proof of Stake (PoS)
Block size	Amount of transactions fitting into a block
IoT Parameters	
Mobility model	Random Waypoint Model (RWM)
Traffic type	Constant Bit Rate (CBR)
Number of IoT devices	100
Simulation time	500 s
Simulation area	$3000\ m \times 3000\ m$
Data rate	10 $Mbps$
Transmitted packet size	$128 - 1024\ B$
Initial energy value	$10 - 12\ J$
Initial trust value	5 J

B. EXPERIMENTAL SETUP

In the following, we provide the details of the simulation environment exploited for the implementation of the proposed framework. As mentioned before, we have leveraged the Mininet-WiFi for the emulation of (software-defined) network topologies, Ethereum as a Blockchain, and OpenFlow-based rules for the realization of the SDN routing capability. All the experiments are run on a hardware architecture with Intel(R) Core(TM) i7 CPU @ 2.50 GHz and 16 GB RAM, with Ubuntu as the operating system.

Table 2 reports a comprehensive list of all the other simulation parameters grouped based on the reference technology (i.e. SDN, Blockchain, and IoT). Particularly, Wireshark (backed by Python scripts) is used to capture and analyze the packets generated by the devices constituting the IoT-SDN network under test. In detail, the IoT environment is simulated for 500 s, with 100 IoT devices dislocated in an area of $3000\ m \times 3000\ m$ and following a Random Waypoint Model [57] of mobility. Each device transmits packets having sizes comprised between 128 B and 1024 B with a data rate of 10 Mbps (constant bit rate), while their initial energy and trust value is set to $10 - 12\ J$ and 5 J, respectively.

Regarding the cluster-head selection algorithm, we evaluate its performance in terms of energy and end-to-end delay required for data transmission between the CH and base station. Additionally, the proposed Blockchain-enabled SDN-IoT architecture is analyzed considering (i) the average throughput (i.e. the amount of transactions among IoT devices in the SDN), (ii) the total energy consumption (depending on the number and energy consumption of network transactions, SDN controllers, and IoT devices), and (iii) the gas consumption (i.e. the amount of computational effort required to execute a certain operation on the Blockchain).

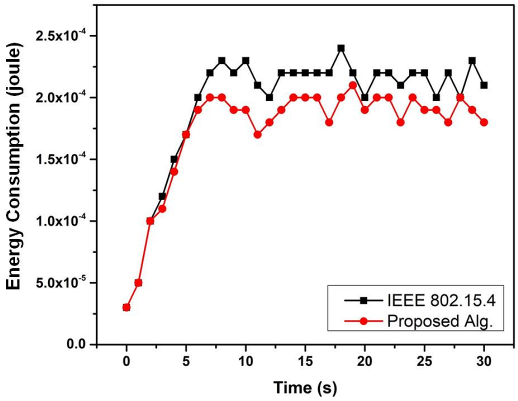


FIGURE 6. Comparison of energy consumption of the proposed cluster-head selection algorithm with the IEEE 802.15.4 protocol at different simulation times. Note that the x-axis is in log-scale.

VIII. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed architecture in terms of different evaluation parameters discussion in Sec. VII-B. Firstly, in Sec. VIII-A, we investigate the performance of the proposed cluster-head selection algorithm in terms of both energy consumption and delay, and compare it with the IEEE 802.15.4 baseline [58]. Then, in Sec. VIII-B we assess the performance of the overall proposed architecture considering the average throughput, energy consumption, and gas consumption, and comparing our proposal with the Blockchain Fundamental (BCF) [46], namely the classical Blockchain that leverages hashing and PoW.

A. CLUSTER-HEAD SELECTION ALGORITHM PERFORMANCE

Energy Consumption

Routing devices—generally, networking components—consume a large amount of energy during data transmission. Specifically, the energy consumption is proportional to the amount of data transmitted by the device (i.e. as many bits transmitted the more energy the device consumes). By leveraging the idea of clustering and employing the CHs for the transmission, the energy consumption can be reduced. The more efficient the cluster-head selection algorithm is, the more effective the clustering technique. To evaluate the energy consumption of the proposed cluster-head selection technique (see Alg. 1), we compare it with the IEEE 802.15.4 protocol [58]. We have simulated both algorithms for 30 seconds and the outcome is shown in Fig. 6. It can be noticed that our proposed algorithm can efficiently select the CHs and has a lower energy consumption than the

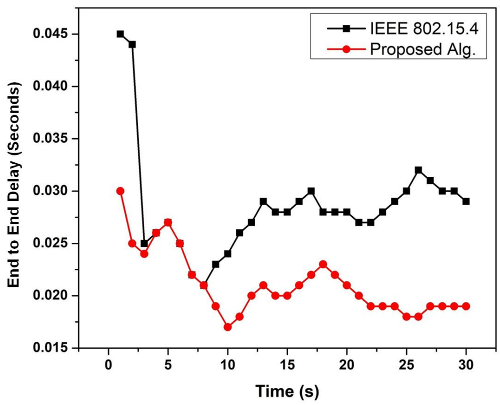


FIGURE 7. End-to-end delay of the proposed cluster-head selection algorithm and IEEE 802.15.4 protocol at different simulation times. Note that the x-axis is in log-scale.

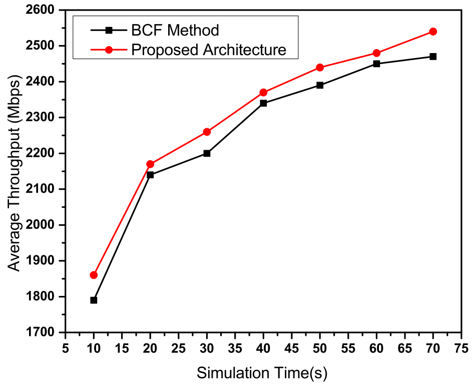


FIGURE 8. Comparison of the average throughput of proposed architecture with the BCF baseline.

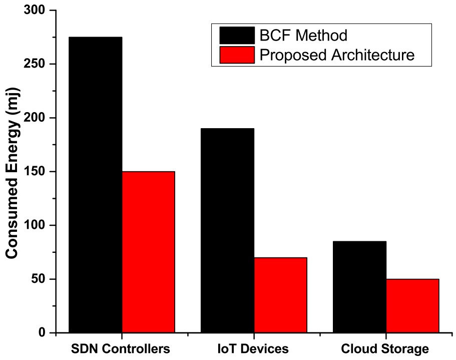


FIGURE 9. Comparison of the energy consumption of the proposed architecture with the BCF baseline.

IEEE 802.15.4 protocol. Also, although both algorithms have similar energy-utilization profiles, the proposed technique has a higher efficiency in energy utilization with increasing simulation time.

End-to-End Delay Analysis

Seeing as how IoT applications are used in real-time systems, it is very crucial to perform all operations in the shortest possible time. Hence, the head of each cluster should be selected very efficiently. To face this issue, our proposed algorithm selects the CHs within a short time frame as we consider the energy level of the sensors according to the G_{dist} distance metric. A node is then marked when it is selected as the CH, or it is associated with another head. Consequently, each node is scanned for cluster-head selection only once.

Taking into account the time a data packet spends for network transmission, we should also investigate this end-to-end delay being dependent on the CH selected. Indeed, the CH should be chosen so that end-to-end delay is minimized during the cluster-head selection process. The curves in Fig. 7 depict the end-to-end delay vs. the elapsed simulation time when both the proposed cluster-selection algorithm and IEEE 802.15.4 protocol are run for 30 seconds. We can notice that the end-to-end delay of both approaches converges with the simulation time, while the proposed algorithm always shows lower end-to-end delay than the IEEE 802.15.4 protocol. Consequently, our proposal presents suitable performance to properly select the CHs to guarantee efficient communication among the routing devices.

B. OVERALL ARCHITECTURE PERFORMANCE

Average Throughput Analysis

Figure 8 depicts the average throughput (in Mbps) of the proposed architecture with respect to simulation time (in seconds), and compares it with the BCF baseline. We can notice that the optimized proposed solution always outperforms BCF over the entire observation period, yet both show increasing trends with time. Interestingly, at first, the average throughput assumes similar values, but the difference becomes sharper as the time grows. Indeed, capitalizing on the implementation of the optimized algorithm for the selection of CHs among IoT devices—along with the security and consistency attained in the SDN via the integration of the Blockchain—our architecture can effectively reduce processing overhead and thus reach better results.

Energy Consumption

As for cluster-head selection, energy consumption is one of the key factors to be managed and optimized in the Blockchain-enabled SDN-IoT architecture. Figure 9 compares the dissipation of energy between our architecture and the BCF method by dividing the energy consumption between three contributing components (i.e. SDN controllers, IoT devices, and Cloud storage). Overall, our proposal outperforms the BCF method that is not able to take into account

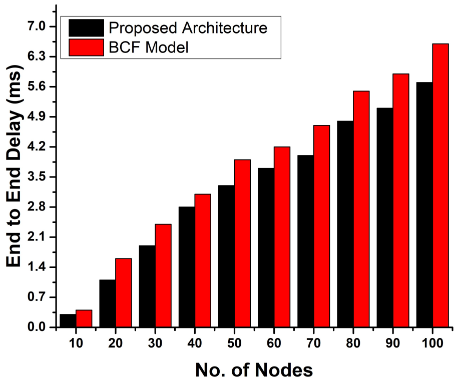


FIGURE 10. Comparison of the overall end-to-end delay (vs. number of nodes) of the proposed architecture with the BCF baseline.

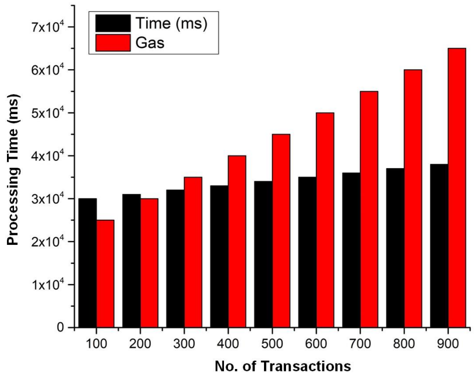


FIGURE 11. Gas consumption with respect to processing time when varying the number of transactions.

the limitations of IoT devices introducing overhead in energy utilization. Conversely, we can efficiently transmit packets using optimized routing paths between the defined clusters of IoT devices. Consequently, the SDN controllers in our architecture consume approximately 50% less energy as opposed to BCF. Also, regarding IoT devices, we can observe that the energy usage of BCF is more than $3\times$ higher than ours (i.e. $\approx 190\ mJ$ vs. $\approx 60\ mJ$). Finally, cloud storage presents the least energy utilization compared to the other two components (i.e. $\approx 50\ mJ$). Nevertheless, also in this case, the BCF method consumes about twice our amount of energy.

Overall End-to-end Delay Analysis

Figure 10 shows the performance in terms of overall end-to-end delay (in ms) by comparing the proposed architecture with the BCF baseline. In detail, we compute the overall end-to-end delay in terms of workloads and time delay by varying the number of nodes. Notably, with a small number of nodes (i.e. ≤ 10) the end-to-end delay of our proposal and BCF assumes similar values. Then, increasing the number of nodes, both end-to-end delays show a linear increasing trend, with our proposal constantly outperforming the BCF method. This result proves the practicality of the proposed architecture that can effectively reduce the end-to-end delay with respect to the existing baseline.

Gas Consumption

Finally, in Fig. 11, we investigate the total gas consumption varying the number of network transactions and comparing it with the processing time (in milliseconds) of each transaction (i.e. the time needed by an SDN controller to process and respond to a request). With a lower number of transactions (i.e. < 400), the gas consumption and processing time assume similar values (up to $\approx 30\ s$). Then, augmenting the transactions, the processing time remains constant, while gas consumption has a linear increase with their number. This result confirms that our architecture is sufficiently scalable since the processing time an SDN controller requires to serve a request maintains lower (and thus suitable) when compared to the gas needed to complete a transaction on the Blockchain. Thus, our proposal is able to combine high safety (provided by Blockchain technology) with efficiency (provided by the optimized SDN-IoT clustered architecture).

IX. DISCUSSION

In the present section, we discuss the outcomes of this research as lessons learned and corresponding open challenges.

Impact of the Research

To improve the real-time experience of IoT applications, time constraints should be taken into account and overall operational time should be optimized. Indeed, a shorter time results in an improved utilization of integrated technologies (i.e. IoT, SDN, and Blockchain). However, managing the resources is extremely challenging with an increasing number of IoT devices. To face this challenge, in the present work, we have designed an optimized Blockchain-SDN framework to manage the IoT resources with a faster (in terms of both time needed to scan the cluster nodes and end-to-end delay) and more energy-efficient cluster-head selection algorithm, which contributes a dimension to the IoT field. Moreover, SDN provides the programmability and flexibility to manage the devices and control them.

Going into details of performed analyses, the present paper has sought to answer the three research questions we have outlined in Sec. I.

A1. Distributed SDN-IoT Ecosystem: We have proposed a layered architecture that aims to optimize IoT-resource management via a distributed Blockchain-based SDN framework. Our architecture is organized in three distinct layers associated with just as many environments and takes advantage of multiple SDN controllers that dynamically and efficiently manage data sent by IoT devices (properly organized using clustering techniques).

A2. Cluster Head Selection: We have devised a novel CH-selection algorithm in the IoT layer of the proposed architecture that takes into account the residual energy of sensor nodes and their distance. We have carefully described the algorithm as a sequence of operational steps and we have proved that it is able to outperform the considered baseline in terms of energy consumption and end-to-end delay.

A3. Distributed SDN Platform Security: We have first discussed most-common SDN vulnerabilities by considering different types of attack (i.e. overloading attack, topology poisoning attack, saturation attack, and DDoS) and describing the threat model employed in the proposed architecture. Possible countermeasures for attack detection and mitigation have been also described. Then, we have shown how cloud-enabled Blockchain can be integrated into our architecture to guarantee the consistency of both data and control layer of the SDN environment. Indeed, Blockchain allows both the management of flow rules and the detection of flow-rule violations.

Limitations and Open Challenges

Unfortunately, deploying the proposed layered architecture in the real world is rather complex. Therefore, we have performed extensive simulation (via virtualization) to evaluate its performance. A first possible limitation of our setup is that we have simulated a system with a limited number of nodes and sensor information. Furthermore, we have not taken into account a mobility scenario in which nodes' position changes from time to time. Indeed, we assume that the nodes remain fixed during clustering and CH selection. Thus, once the CHs are selected, the architecture works with the chosen set of CHs (and nodes) until the whole procedure is executed again. Also, we have emphasized only two (most impacting) parameters, but there are other factors that could be effective to perform architecture management and CH selection. However, the virtualization needed to simulate our environment is time-consuming. Hence, this issue could be another interesting possible line of investigation for the researchers. Indeed, a real-world implementation is hampered by the fact that our architecture is based on Blockchain, and still now, Blockchain implementation remains a challenge in the real world.

X. CONCLUSIONS

Blockchain-enabled software-defined IoT ecosystems suffer from immature workflow definitions in the developing stage and also from lack of resources to deploy and manage this ecosystem properly. Besides, only a limited number of previous researches has investigated and addressed these issues.

Based on these considerations, we have introduced an optimized comprehensive framework for resource management in the Blockchain-enabled software-defined IoT ecosystem, encompassing a novel and efficient cluster-head selection algorithm and a distributed flow-rule verification technique that guarantees consistency and security to the network.

Additionally, our framework—deployed in a layered architecture—maintains multiple homogeneous SDN controllers that are able to enhance the availability, confidentiality, and integrity in the IoT ecosystem. Specifically, our cluster-head selection algorithm implements an efficient procedure for selecting the cluster heads with optimized energy consumption, required in a constrained environment with limited resources. Furthermore, the experimental evaluation (whose implementation scenario is also described) has demonstrated that our proposal outperforms the considered baseline on both energy utilization and end-to-end delay. Overall, the Blockchain-enabled SDN-IoT architecture attains better performance (in terms of average throughput, energy utilization, and overall end-to-end delay) compared to a classical Blockchain. Also, the SDN controllers show a processing time suitable compared to the gas consumed by transactions performed on the Ethereum Blockchain.

In the future, we plan to enhance the features of the framework and deploy a real-world large-scale scenario of the proposed architecture. Additionally, we will employ our cluster-head selection algorithm in a mobility scenario to evaluate its responsiveness and adaptability to changes in the environment.

REFERENCES

[1] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[2] P. Megyesi, A. Botta, G. Aceto, A. Pescapé, and S. Molnár, "Challenges and solution for measuring available bandwidth in software defined networks," *Computer Communications*, vol. 99, pp. 48–61, 2017.

[3] A. Al-Hayajneh, Z. A. Bhuiyan, and I. McAndrew, "Improving internet of things (iot) security with software-defined networking (sdn)," *Computers*, vol. 9, no. 1, p. 8, 2020.

[4] Z. Shao, X. Zhu, A. M. Chikuvanyanga, and H. Zhu, "Blockchain-based sdn security guaranteeing algorithm and analysis model," in *International Conference on Wireless and Satellite Systems*. Springer, 2019, pp. 348–362.

[5] C.-Y. Shieh and T. W. F. Chou, "Methods and systems for improving analytics in distributed networks," Jan. 29 2019, uS Patent 10,193,929.

[6] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[7] K. Košt'ál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak, "Management and monitoring of iot devices using blockchain," *Sensors*, vol. 19, no. 4, p. 856, 2019.

[8] M. P. Hossain, M. Khaled, S. A. Saju, S. Roy, M. Biswas, and M. A. Rahaman, "Vehicle registration and information management using blockchain based distributed ledger from bangladesh perspective," in *2020 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2020, pp. 900–903.

[9] A. Khatoon, P. Verma, J. Southernwood, B. Massey, and P. Corcoran, "Blockchain in energy efficiency: Potential applications and benefits," *Energies*, vol. 12, no. 17, p. 3317, 2019.

[10] A. A. Abbasi, A. Abbasi, S. Shamshirband, A. T. Chronopoulos, V. Persico and A. Pescapè, "Software-Defined Cloud Computing: A Systematic Review on Latest Trends and Developments," *IEEE Access*, vol. 7, pp. 93294-93314, 2019.

[11] A. Rahman, M. J. Islam, M. S. I. Khan, S. Kabir, A. I. Pritom, and M. R. Karim, "Block-sdotcloud: Enhancing security of cloud storage through blockchain-based sdn in iot network," *In progress: 2nd International*

Conference on Sustainable Technologies for Industry 4.0 (STI 2020), IEEE Xplore, 2020.

[12] M. J. Islam, M. Mahin, A. Khatun, S. Roy, S. Kabir, and B. C. Debnath, “A comprehensive data security and forensic investigation framework for cloud-iot ecosystem,” *GUB Journal of Science and Engineering*, vol. 4, 2019.

[13] Cisco,“White paper: Cisco Annual Internet Report (2018–2023),”2020, online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[14] S.-K. Kim, U.-M. Kim, and J.-H. Huh, “A study on improvement of blockchain application to overcome vulnerability of iot multiplatform security,” *Energies*, vol. 12, no. 3, p. 402, 2019.

[15] M D’Arienzo, A. Pescapè, and G. Ventre, “Dynamic Service Management in Heterogeneous Networks,” *Journal of Network and Systems Management*, vol. 12, p. 349–370, 2004.

[16] B. K. Mukherjee, M. S. I. Pappu, M. J. Islam, and U. K. Acharjee, “An SDN based Distributed IoT Network with NFV Implementation for Smart Cities,” *2nd International Conference on Cyber Security and Computer Science (ICONCS-2020)*, Springer, 2020.

[17] A. Muthanna, A. A Ateya, A. Khakimov, I. Gudkova, A. Abuarqoub, K. Samouylov, and A. Koucheryavy, “Secure and reliable iot networks using fog computing with software-defined networking and blockchain,” *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 15, 2019.

[18] M. J. Islam, M. Mahin, S. Roy, B. C. Debnath, and A. 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)*. IEEE, 2019, pp. 1–6.

[19] T. Takenaka, Y. Yamamoto, K. Fukuda, A. Kimura, and K. Ueda, “Enhancing products and services using smart appliance networks,” *CIRP Annals*, vol. 65, no. 1, pp. 397–400, 2016.

[20] F. Al Shuhaimi, M. Jose, and A. V. Singh, “Software defined network as solution to overcome security challenges in iot,” in *Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2016 5th International Conference on*. IEEE, 2016, pp. 491–496.

[21] A. Rahman, M. J. Islam, Z. Rahman, M. M. Reza, A. Anwar, M. A. P. Mahmud, M. K. Nasir, and R. M. Noor, “Distb-condo: Distributed blockchain-based iot-sdn model for smart condominium,” *IEEE Access*, vol. 8, pp. 209 594–209 609, 2020.

[22] A. Rahman, M. J. Islam, F. A. Sunny, and M. K. Nasir, “DistBlockSDN: A Distributed Secure Blockchain based SDN-IoT Architecture with NFV Implementation for Smart Cities,” *In Press: International Conference on Innovation in Engineering and Technology (ICIET)*, vol. 23, p. 24, IEEE, 2019.

[23] A. Rahman, M. K. Nasir, Z. Rahman, A. Mosavi, S. S., and B. Minaei-Bidgoli, “Distblockbuilding: A distributed blockchain-based sdn-iot network for smart building management,” *IEEE Access*, vol. 8, pp. 140 008–140 018, 2020.

[24] A. Rahman, U. Sara, D. Kundu, S. Islam, M. J. Islam, M. Hasan, Z. Rahman, and M. K. Nasir, “Distb-sdoindustry: Enhancing security in industry 4.0 services based on distributed blockchain through software defined networking-iot enabled architecture,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110980

[25] M. J. Islam, A. Rahman, S. Kabir, A. Khatun, A. I. Pritom, and M. Zaman, “Sdot-nfv: Enhancing a distributed sdn-iot architecture security with nfv implementation for smart city,” 2020.

[26] S. A. A. Kazmi, M. K. Shahzad, A. Z. Khan, and D. R. Shin, “Smart distribution networks: A review of modern distribution concepts from a planning perspective,” *Energies*, vol. 10, no. 4, p. 501, 2017.

[27] Y. Huang, J. Tan, and Y.-C. Liang, “Wireless big data: transforming heterogeneous networks to smart networks,” *Journal of Communications and Information Networks*, vol. 2, no. 1, pp. 19–32, 2017.

[28] S. Chakrabarty and D. W. Engels, “A secure iot architecture for smart cities,” in *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE, 2016, pp. 812–813.

[29] A. Kumar *et al.*, “Energy efficient clustering algorithm for wireless sensor network,” Ph.D. dissertation, Lovely Professional University, 2017.

[30] K. J. C. Angel and E. G. D. P. Raj, “Eeeca: Enhanced energy efficient clustering algorithm for mobile wireless sensor networks,” in *2017 World Congress on Computing and Communication Technologies (WCCCT)*. IEEE, 2017, pp. 267–270.

[31] A. B. Guiloufi, N. Nasri, and A. Kachouri, “Energy-efficient clustering algorithms for fixed and mobile Wireless Sensor Networks,” *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, 2014, pp. 735–738.

[32] A. Al-Baz and A. El-Sayed, “A new algorithm for cluster head selection in leach protocol for wireless sensor networks,” *International journal of communication systems*, vol. 31, no. 1, p. e3407, 2018.

[33] L. Zhao, S. Qu, and Y. Yi, “A modified cluster-head selection algorithm in wireless sensor networks based on leach,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 287, 2018.

[34] S. Murugaanandam and V. Ganapathy, “Reliability-based cluster head selection methodology using fuzzy logic for performance improvement in wsns,” *IEEE Access*, vol. 7, pp. 87 357–87 368, 2019.

[35] A. Shafique, G. Cao, M. Aslam, M. Asad, and D. Ye, “Application-aware sdn-based iterative reconfigurable routing protocol for internet of things (iot),” *Sensors*, vol. 20, no. 12, p. 3521, 2020.

[36] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, “Residual energy-based cluster-head selection in wsns for iot application,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5132–5139, 2019.

[37] M. Shyjith, C. Maheswaran, and V. Reshma, “Optimized and dynamic selection of cluster head using energy efficient routing protocol in wsn,” *Wireless Personal Communications*, pp. 1–23, 2020.

[38] A. Ouhab, T. Abreu, H. Slimani, and A. Mellouk, “Energy-efficient clustering and routing algorithm for large-scale sdn-based iot monitoring,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[39] I. Ullah and H. Y. Youn, “Efficient data aggregation with node clustering and extreme learning machine for wsn,” *The Journal of Supercomputing*, pp. 1–27, 2020.

[40] S. N. Matheu, A. Robles Enciso, A. Molina Zarca, D. Garcia-Carrillo, J. L. Hernández-Ramos, J. Bernal Bernabe, and A. F. Skarmeta, “Security architecture for defining and enforcing security profiles in dlt/sdn-based iot systems,” *Sensors*, vol. 20, no. 7, p. 1882, 2020.

[41] A. Molina Zarca, D. Garcia-Carrillo, J. Bernal Bernabe, J. Ortiz, R. Marin-Perez, and A. Skarmeta, “Enabling virtual aaa management in sdn-based iot networks,” *Sensors*, vol. 19, no. 2, p. 295, 2019.

[42] M. Conti, P. Kaliyar, and C. Lal, “Censor: Cloud-enabled secure iot architecture over sdn paradigm,” *Concurrency and Computation: Practice and Experience*, vol. 31, no. 8, p. e4978, 2019.

[43] A. Abdelaziz, A. T. Fong, A. Gani, U. Garba, S. Khan, A. Akhunzada, H. Talebian, and K.-K. R. Choo, “Distributed controller clustering in software defined networks,” *PloS one*, vol. 12, no. 4, p. e0174715, 2017.

[44] Y. Liu, Y. Kuang, Y. Xiao, and G. Xu, “Sdn-based data transfer security for internet of things,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 257–268, 2017.

[45] Alberto Dainotti, Antonio Pescapè, Giorgio Ventre: A cascade architecture for DoS attacks detection based on the wavelet transform. J. Comput. Secur. 17(6): 945-968 (2009)

[46] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K. R. Choo, “An energy-efficient sdn controller architecture for iot networks with blockchain-based security,” *IEEE Transactions on Services Computing*, 2020.

[47] R. Chaudhary, A. Jindal, G. S. Aujla, S. Aggarwal, N. Kumar, and K.-K. R. Choo, “Best: Blockchain-based secure energy trading in sdn-enabled intelligent transportation system,” *Computers & Security*, vol. 85, pp. 288–299, 2019.

[48] Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, “Cochain-sc: An intra- and inter-domain ddos mitigation scheme based on blockchain using sdn and smart contract,” *IEEE Access*, vol. 7, pp. 98 893–98 907, 2019.

[49] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, “Blockchain technologies for the internet of things: Research issues and challenges,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2018.

[50] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, and J. Shen, “Defending against flow table overloading attack in software-defined networks,” *IEEE Transactions on Services Computing*, vol. 12, no. 2, pp. 231–246, mar 2019.

[51] S. Hong, L. Xu, H. Wang, and G. Gu, “Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures,” 2015. [Online]. Available: http://dx.doi.org/10.14722/ndss.2015.23283

[52] M. Ambrosin, M. Conti, F. De Gaspari, and R. Poovendran, “LineSwitch: Tackling Control Plane Saturation Attacks in Software-Defined Network-

ing," *IEEE/ACM Transactions on Networking*, vol 25, no. 2, pp. 1206–1219, 2016.

[53] J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo, and D. Zhu, "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," *IEEE Access*, vol. 8, pp. 155 859–155 872, 2020.

[54] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[55] P.-W. Chi, C.-T. Kuo, J.-W. Guo, and C.-L. Lei, "How to detect a compromised sdn switch," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2015, pp. 1–6.

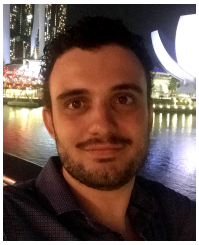
[56] S. Shamshirband, M. Fathi, A.T. Chronopoulos, A. Montieri, F. Palumbo, and A. Pescapè, "Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues," *Journal of Information Security and Applications*, vol. 55, pp. 2214–2126, 2020.

[57] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.

[58] N. Kushalnagar, G. Montenegro, C. Schumacher *et al.*, "Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals," 2007.

ANICHUR RAHMAN received the B.Sc. and M.Sc degree in Computer Science and Engineering from Mawlana Bhashani Science and Technology University (MBSTU), Tangail, Bangladesh in 2017 and 2020 respectively. Currently, he is working as a Lecturer at Computer Science and Engineering (CSE), National Institute of Textile Engineering and Research (NITER), Savar, Dhaka, Bangladesh since January 2020 to present. His research interests include Internet of Things (IoT), Blockchain (BC), Software Defined Networking (SDN), Machine Learning, 5G, Industry 4.0 and Data Science.

MD. JAHIDUL ISLAM received the B.Sc. and M.Sc. degrees in Computer Science and Engineering from Jagannath University (Jnu), Dhaka, in 2015 and 2017 respectively. Currently, he is working as a Lecturer and Program Coordinator (Day) at Computer Science and Engineering (CSE), Green University of Bangladesh (GUB), Dhaka, Bangladesh since May 2017 to present. He is a member of Computing and Communication and Human-Computer Interaction (HCI) research groups, CSE, GUB. His research interests include Internet of Things (IoT), Blockchain, Network Function Virtualization (NFV), Software Defined Networking (SDN), 5G, Industry 4.0, Machine Learning, HCI, and Wireless Mesh Networking (WMN).



ANTONIO MONTIERI (GSM'18) is a Postdoctoral Researcher at DIETI of the University of Napoli Federico II. He has received his Ph.D. degree in Information Technology and Electrical Engineering in April 2020 from the same University. His work concerns network measurements, (encrypted and mobile) traffic classification, traffic modeling and prediction, and monitoring of cloud network performance. Antonio has co-authored 23 papers in international journals and conference proceedings.

MOSTOFA KAMAL NASIR Professor of Computer Science and Engineering of Mawlana Bhashani Science and Technology University, Tangail, Bangladesh. He has completed his PhD from University of Malaya, Kuala Lumpur, Malaysia in the field of Mobile Adhoc Technology in 2016. Before that he has completed his BSc and MSc in Computer Science and Engineering from Jahangirnagar University, Bangladesh. His current research interest include VANET, IoT, SDN and WSN.

MD. MAHFUZ REZA received his B.Sc.(Engg.) in Computer Science and Engineering (CSE) from Mawlana Bhashani Science and Technology University (MBSTU), Tangail, Bangladesh in 2011. Also, he received M.Sc.(Engg.) in CSE from MBSTU in 2015. Currently, he is an Associate Professor of CSE, MBSTU. His current research interests in the areas of data sciences, machine learning, sensor networks, IoT, cryptography and network security.

SHAHAB S. BAND received the M.Sc. degree in artificial intelligence from Iran, and the Ph.D. degree in computer science from the University of Malaya (UM), Malaysia, in 2014. He was an Adjunct Assistant Professor with the Department of Computer Science, Iran University of Science and Technology. He also severed as a Senior Lecturer with UM, Malaysia, and with Islamic Azad University, Iran. He participated in many research programs within the Center of Big Data Analysis, IUST and IAU. He has been associated with young researchers and elite club, since 2009. He supervised or co-supervised undergraduate and postgraduate students (master's and Ph.D.) by research and training. He has also authored, or coauthored papers published in IF journals and attended to high-rank A and B conferences. He is an Associate Editor, a Guest Editor, and a Reviewer of high-quality journals and conferences. He is a professional member of the ACM.

ANTONIO PESCAPÉ (SM'09) is a Full Professor of computer engineering at the University of Napoli Federico II. His work focuses on measurement, monitoring, and analysis of the Internet. He has co-authored more than 200 conference and journal papers, he is the recipient of a number of research awards. Also, he has served as an independent reviewer/evaluator of research projects/project proposals co-funded by a number of governments and agencies.

MAHEDI HASAN received M.Sc. in Database Technology from Saint Petersburg State University, Russia in 2016 and the B.Sc. (Engg.) degree in Information and Communication Technology from Mawlana Bhashani Science and Technology University (MBSTU), Tangail, Bangladesh in 2012. Currently, he is working as a Lecturer at Computer Science and Engineering (CSE), National Institute of Textile Engineering and Research (NITER), Savar, Dhaka, Bangladesh since March 2020 to present. His research interests are Data Science, Machine Learning (ML) and Natural Language Processing (NLP).

MEHDI SOOKHAK (SM'09) received the Ph.D. degree in computer science, major in information security, from the University of Malaya (UM), in 2015. He was an Active Researcher with the Center of Mobile Cloud Computing Research (C4MCCR), UM. From 2016 to 2017, he was with Carleton University, Canada, as a Postdoctoral Fellow. He is currently an Assistant Professor of cybersecurity with Illinois State University, Normal, IL, USA. He has authored more than 40 articles in high ranking journals and conferences. He is an Editor of several ISI journals and chair of several conferences. His areas of interest include cloud and mobile cloud computing, fog computing, vehicular cloud computing, the IoT and smart cities, computation outsourcing, access control, network security, wireless sensor & mobile Ad Hoc network (architectures, protocols, security, and algorithms), big data security and analytic, distributed systems, and cryptography and information security.

AMIR MOSAVI is an Alexander von Humboldt research fellow for big data, IoT, and machine learning. He is a senior research fellow at Oxford Brookes University. Amir completed his graduate studies at London Kingston University, UK, and received his Ph.D. in applied informatics. He is a data scientist for climate change, sustainability, and hazard prediction. He is the recipient of the Green-Talent Award, UNESCO Young Scientist Award, ERCIM Alain Bensoussan Fellowship Award, Campus France Fellowship Award, Campus Hungary Fellowship Award, and Endeavour-Australia Leadership.