# Software-Defined Networks (SDNs) and Internet of Things (IoTs): A Qualitative Prediction for 2020

Sahrish Khan Tayyaba
Department of Computer Science,
COMSATS Institute of Information Technology,
Islamabad, Pakistan

Naila Sher Afzal Khan
University of Management Sciences and Information Technology,
Kotli, AJK

Wajeeha Naeem
Department of Computer Science,
COMSATS Institute of Information Technology
Islamabad, Pakistan

Munam Ali Shah
Department of Computer Science,
COMSATS Institute of Information Technology
Islamabad, Pakistan

Yousra Asim
Department of Computer Science,
COMSATS Institute of Information Technology
Islamabad, Pakistan

Muhammad Kamran
Department of Distance Continuing & Computer Education,
University of Sindh,
Hyderabad, Pakistan

***Abstract*—The Internet of Things (IoT) is imminent technology grabbing industries and research attention with a fast stride. Currently, more than 15 billion devices are connected to the Internet and this number is expected to reach up to 50 billion by 2020. The data generated by these IoT devices are immensely high, creating resource allocation, flow management and security jeopardises in the IoT network. Programmability and centralised control are considered an alternative solution to address IoT issues. On the other hand, a Software Define Network (SDN) provides a centralised and programmable control and management for the underlying network without changing existing network architecture. This paper surveys the state of the art on the IoT integration with the SDN. A comprehensive review and the generalised solutions over the period 2010-2016 is presented for the different communication domains. Furthermore, a critical review of the IoT and the SDN technologies, current trends in research and the futuristic contributing factors form part of the paper. The comparative analysis of the existing solutions of SDN based IoT implementation provides an easy and concise view of the emerging trends. Lastly, the paper predicts the future and presents a qualitative view of the world in 2020.**

***Keywords*—SDN; IoT; Integration of SDN-IoT; WSN; LTE; M2M communication; NFV**

## I. INTRODUCTION

The emergence of new technologies and communication networks offer new connectivity scenarios among every physical object. Machine-to-Machine (M2M), Device-to-Device (D2D), Vehicle-to-Vehicle (V2V), wireless sensor network, actuators, smartphone, embedded devices and even connections among infrastructures are developing new connectivity scenarios. Moreover, these devices will be allegedly connected to the Internet and will ultimately create a heterogeneous system of interconnected objects; called the Internet of Thing (IoT), and in broader sense Internet of Everything (IoE) [1]. The IoT devices are generally sensor node, actuator, RFID tags and wireless communicating devices connected to the Internet in a smart environment. The IoT devices are capable of observing, analysing and taking intelligent decisions based on collected information from the surroundings and manipulation of the underlying network. The IoT devices are deployed according to the customised task with specific applications; forming a domain specific IoTs network. This domain specific applications and service attribute a horizontal view of the IoT network such as appliances and applications for smart home management, smart health care unit implanted on the body or wearable sensors for health monitoring. The domain-based services can leverage the benefits of pervasive and ubiquitous computing through the independent services horizontal platform.

With the immense increase in IoT devices huge amount of data is generated and collected which impede monitoring, management, controlling and securing IoT devices in a heterogeneous network and become a critical issue for researchers and developers. Traditional network does not completely support heterogeneity, which limits IoT benefits full realisation. In addition, the services demand and customers require fast development and deployment that is still an issue in a traditional network. The innovation in the legacy network is very slow due to the proprietary nature of devices. Therefore, a change in the traditional network infrastructure and devices is mandatory to realise full benefits of IoTs. IoT can leverage full benefits from the integrated architecture of such technologies. The most attracted technologies in this domain are Software Defined Networking (SDN), and Network Function Virtualization (NFV).

SDN is an emerging technology that can meet the need of current IoT requirements of heterogeneity and flexibility. It provides a centralised control and global view of the whole network. SDN decouple the control functionality from the forwarding plane and program network service sitting above the controller (control Plane). The centralised management facilitates optimisation and configuration of a network in an efficient and automated manner and provides interoperability among heterogeneous IoT network. This control plane centralization can provide a secure architecture for IoT

network, e.g., smart home security applications prevent unauthorised user access of the smart appliance etc. IoT is growing with a very fast stride that new trends and technologies, protocols, architecture, management, and security solutions in the context of IoT are formulated within a short period. There is a research gap in addressing the IoT integration with different networking solutions especially, leveraging the benefits of SDN.

In this paper, we highlight different studies which provide SDN based solutions for IoT technologies. We survey the literature over the period 2010-2016, by focusing the attention on different aspects of the IoT merger with the SDN. The organisation of this paper is as follows. Section II provides some background of the IoT and the SDN and architecture of two contributing domains, i.e., SDN and the IoT and the protocols for the SDN. In section III, a comprehensive literature is provided for the existing solution of the SDN and the IoT integration. Section IV provides a detailed review of the existing solution, providing a comparative analytics of the existing integration solutions. In section V, market and research trends and a qualitative prediction for 2020 are given. Section VI concludes the study.

## II. Background Related Studies

### A. Background

The use of computing devices and communication technologies are growing exponentially with the decline in cost and size of hardware and software. Vendors and organisations are digging new domain in search of finding new ways of flexible computing and communication. IoT and SDNs are two complete different communication and network domain whose merger is seeking for benefiting human kinds and developing smart systems. As the IoT implementation expectancy exceeds the limits of traditional network e.g., Virtual Private Network (VPN), the SDN promise to hold the traditional network with new service demands. At this stage, technology shift is highly intention grabbing a task from the researchers and developers in industries and organisations. The two domains and their architecture are totally dissimilar. In this section, an architectural detail of both domains is presented to grab the underlying functionality for the merging of IoT in SDN.

### B. SDN Architecture and protocol

In a traditional network, the devices and the equipment are usually proprietary entities, are physically distributed and control function is hard-coded. The network operator has to do configuration of the individual network device as per service layer agreements (SLAs) and cannot be programmed otherwise. The complexity increases due to the vertical integration of network architecture. The control plane and the data plane are bundled inside the networking devices, reducing flexibility and hindering innovation and evolution of the networking infrastructure. Any change in the network is expensive in term of time, and cost. The cost comes in term of capital expenditure (CAPEX) and operational expenditure (OPEX) [2]. For example, the transition from IPv4 to IPv6, started more than a decade ago and still largely incomplete, bears witness to this challenge, while in fact, IPv6 represented merely a protocol update. To overcome the existing architecture, SDN is considering as the best alternate.

In SDN, the control plane is decoupled from forwarding plane and communication between two planes is done through using Southbound and Northbound APIs. SDN is basically layer architecture consists of three layers 1). Device layer or data plane 2). Control plane and 3). Application layer. The customer needs are abstracted over application layer which is communicated to the controller via Northbound APIs e.g., RESTfull API. The control layer or controller is centralised part of the SDN network and act as a brain of the network. The controller manages the whole network and possesses a global view of the network. All applications/programs run above the controller. Many controllers are in the market from its inception such as ONOS, Open daylight, Floodlight, NOX [3], POX, Trema etc. SDN controller define rule for the incoming flows from the data plane. The controller communicates with the devices in the data plane via Southbound APIs, most common and recognised is OpenFlow (OF). The layered architecture of SDN is shown in Figure 1

SDN do not increase the performance of the network rather it provides flexibility in network configuration and resource management. On the contrary, SDN can lead to performance degradation in case of providing high level of abstraction

#### 1) SDN architecture

SDN is a layered architecture, consisting of three basic layers; application/services layer, controller layer (control plane), and data plane layer called forwarding layer consisting of forwarding devices. These SDN layers communicate with each other via open APIs called Northbound Interface (NI) API and Southbound Interface (SI) API [5]. To identify the different elements of an SDN as clearly as possible, we now present the essential terminology used throughout this work

##### a) SDN architectural components

SDN is a layered architecture, consisting of three basic layers; application/services layer, a controller layer, and data plane layer called forwarding layer consisting of forwarding devices. These SDN layers communicate with each other via open APIs called Northbound Interface (NI) API and Southbound Interface (SI) API [5].

SDN layered components are described to

- *Application layer (AP):* The application plane also called management plane consist of applications that leverage the functions offered by the NI to implement network control and operation logic. Essentially, a management application defines the policies, which are ultimately translated to southbound-specific instructions that program the behaviour of the forwarding devices.

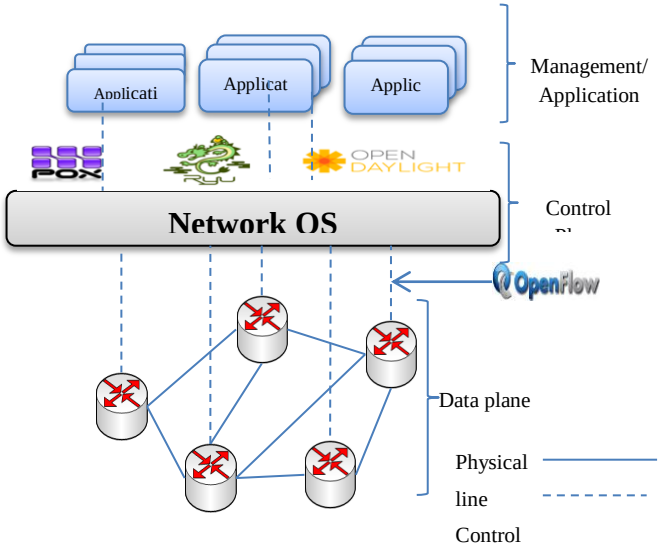


Fig. 1. SDN Architecture

- *Northbound Interface (NI):* The interaction between application AP and control plane is provided through NI. The Network Operating System (NOS) facilitate application developers to coordinate through these NI APIs. Typically, an NI APIs abstracts the low-level instruction sets and implementation of forwarding devices. So far NI APIs is not well studied. Generally, RESTFull APIs are used as an interface between applications and control plane.
- *Control Plane (CP):* Control plane is the decoupled entity from the distributed forwarding devices and logically centralised on a server. CP programs the forwarding devices through southbound interfaces. CP defines rules/instruction set for forwarding devices hence control plane is the ‘‘network brain’’ and all control logic rests in the applications and controllers, which form the control plane. Many SDN controllers are available in the market such as NOX[3], OpenDaylight[5], Ryu[6].
- *Southbound Interface (SI):* Southbound interfaces provide a communication protocol between CP and forwarding device though the SI instruction set. Well established SI protocol help controller in programming forwarding devices and formalise rules for interaction between the two planes (CP & DP). Some examples are OpenFlow [7], Forwarding and Control Elements (ForCES) [8] , Protocol-oblivious forwarding (POF) [9].
- *Forwarding Devices (FD):* Network core devices either software based or hardware based performs fundamental network operations. The forwarding devices act on the basis of rules/instruction set provided by CP/controller on the incoming flow/packets (e.g., forward, drop, rewrite some header). These instructions are defined by southbound interfaces such as OpenFlow [7], ForCES [8] and are installed in the forwarding devices by the SDN controllers implementing the southbound protocols.
- *Data Plane (DP)/Forwarding Plane:* Forwarding devices (routers, switches, gateways etc.) are interconnected through a physical medium such as wireless radio channels or wired cables. And defined a physical interconnection within a network

SDN has many applications in other networks such as in management, configuration and reconfiguration of the network in a flexible manner. SDNs provide a fine-grained control with high quality of services. The SDN controller flexibly manages the flow forwarding state in the data plane (router & switches) by having a global view of the network. SDN controller provides programmability for the data plane. Controller is logically centralised entity but physically distributed [10]. SDN is believed to provide its user with a separate networking slice by utilising the concept of virtualization. NFV is considered as a complementary technology for SDN. SDN utilised the virtual view of the network status and provide different applications based on this virtualized view. NFV can be implemented as an application above the CP. Network functions can be virtualized in NFV. The next generation network architecture is quite dependent on such technologies which can facilitate high data transmission, spectral efficiency, resource allocation and network management for fulfilling growing need of the customer demands. One solution to such demand is the programmability of the network and dynamic allocation of resources, which can be provided by network virtualization. In virtualization, user specific network is called slice, which provides new values to user requirements and applications. In the next section, we will highlight the detailed architecture of IoT network, which is again layered architecture of connecting the physical object with the Internet.

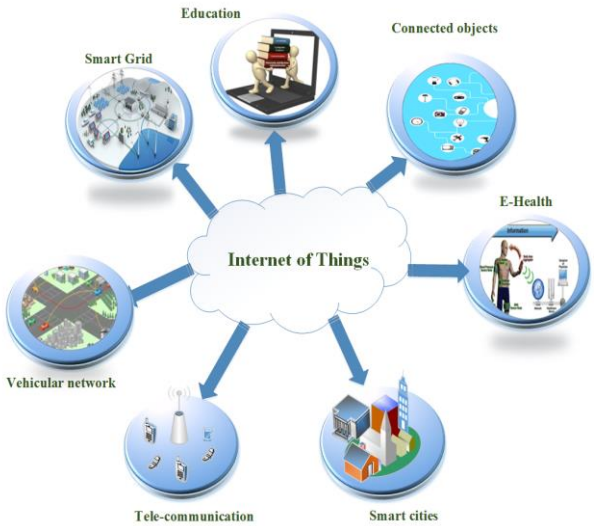


Fig. 2. Overall IoT scenario

*2) IoT architecture*

We are living in the era of connected objects where devices can communicate with the physical world and capable

of taking decisions due to the data analytics. The main factor behind this swift shift is the advancement in the microelectronics, telecommunication networks, and use of Radio Frequency Identification (RFID) tags attached to the physical objects. When these objects are connected to the Internet, they form a network of interconnected objects called IoT. The IoT is simply the point in time when more things or objects are connected to the Internet than people. [11]. As the boundaries of connected objects are not limited to certain technology, diverse ranges of objects connect and communicate with each other using a different communication protocol, resulting in the heterogeneous network as visible in Fig. 2. IoT devices are used to sense, collect, process, infer, transmit, notify, manage, and store data. The IoT helps in building a smart environment. Few examples are home safety and management system, smart electricity monitoring in electricity grids, in-car system from road traffic monitoring to control function and safety measures in advances, health monitoring to smart building automatically controlled heating, venting, and air conditioning (HVAC) systems, security systems, disaster management, weather forecasting etc. are variant domain and provide a powerful control in handling daily life activities. There are billions of devices connected to the heterogeneous network. These entire domains have different architectural details as per the specified functional requirement and still not converged on are not converged on a single reference model [12], which add complexity in the heterogeneity of a network. However, the general architecture of IoT is shown in the Fig. 3

*a) IoT architectural components*

For any network, layered architecture ensures flexibility and capability of invocation of new services in the network, IoT architecture follows layered architecture. Due to varying IoT domain, architecture and contributing components are not converged however most successful IoT architecture is IoT-A [13]. Many other IoT architecture models are also in the market but most common is "four-layer architecture"

- *Perception layer:* Perception layer is physical object layer consisting of sensors, actuator, RFIDs, mobile devices, motes, blue tooth etc. This layer collects the data from the environment and transmits on the edge of the network i.e. gateway or sink.
- *Network layer:* This layer is responsible for transmitting data from physical objects to the gateway/edge of the network for further processing on the collected information. Different transmission technologies contribute to the heterogeneity of IoT such as ZigBee, blue tooth, Wi-Fi etc.
- *Application layer:* This layer deal with the application/services of the user demand by manipulating the information collected from the perception layer and processed in the processing system.

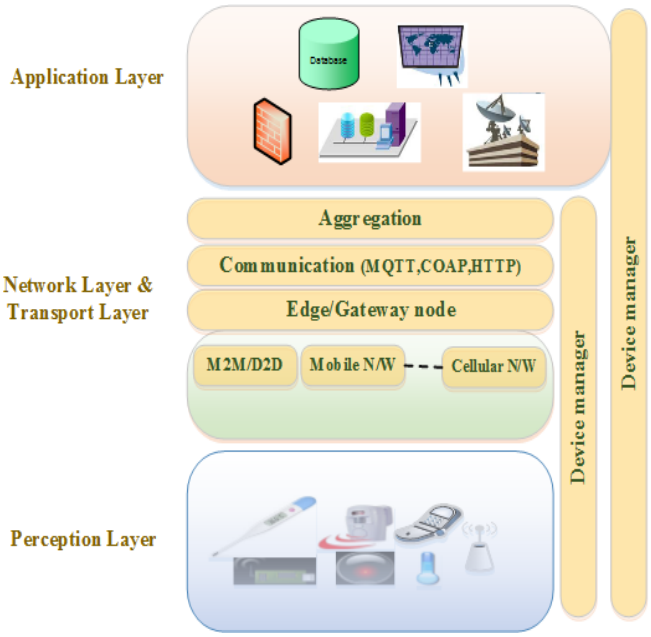


Fig. 3. IoT Architecture

- *Middleware layer:* Different IoT devices in a domain may be different but devices can interact with a compatible/same device. This layer translates the message of one service information without concern for the hardware detail. Middleware layer is associated with service management, addressing and naming of the requested service.

Beside these main layers, there are many components, which play important role in IoT information collection, processing and management. We define these components. *Edge services* component is responsible for delivering information through the Internet. These services may be domain name service, Content Delivery Network, firewall, load balancer etc. *Analytics services* component guide and automates the process of data analysis, discovery, and visualisation. The *Process management services* help in managing the workflow of the information processing and connects devices with their respective services. *Device identity* services identify a user registers service on a device. A*uthentication* service enables the authentication of a registered user with its associated service. *Service Oriented Architecture (SOA)* helps in providing architectural abstraction from the underlying detail and provides required services.

Initially, the Internet was distinctly established over TCP/IP suit and provided support for a large number of the connected computer. However, TCP/IP does not support heterogeneous network. Therefore, the TCP/IP is not suitable for IoTs. Hence, the heterogeneity of connected device in IoT environment is creating unprecedented complexity and functional diversity.

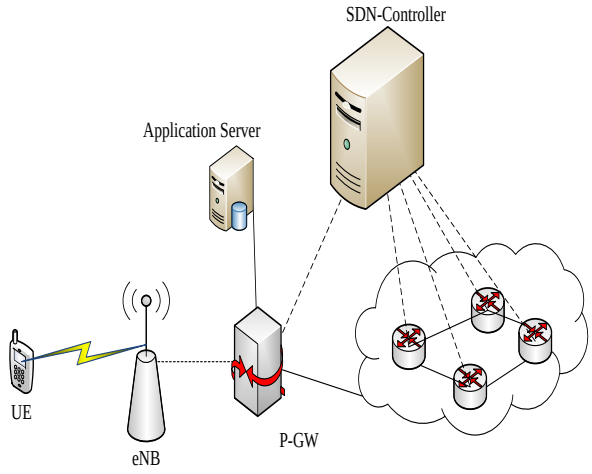


Fig. 4. SDN based LTE architecture

### C. Related Studies

SDNs and IoTs are a hot topic and grabbing the attention of industry and market. Many comprehensive studies are done by the research communities to look into the detail perspective; implications, use cases, and technological demand in both domains. Kreutz et al. [2] present a comprehensive study on SDN. The authors provide a detail and all-inclusive on SDN, SDN evolution from programmable networks to SDN architecture, protocols, application, use case scenarios and future research trends etc. Nunes et al. in [3], discuss the past, present and future of programmable network based on SDN. SDN layered taxonomy is presented in [4]. Network innovation in the context of SDN using OpenFlow is dealt in [5].

An ample survey on IoT is presented by Al-Fuqaha et al. in [6], mentioning every domain of IoT, application, issues and scenarios. Similarly, Xu et al. [7] presents state-of-the-art on IoT. The future industrial perspective of IoT is presented in [8]. The study of IoT applications is done in [9]. The merger of IoT and SDN is also studied in many research articles as in [10] which presents the SDN and virtualization in IoT domain. However, a detailed survey on the integration of IoT in SDN requires attention from the research community.

## III. Literature Review

Since SDN and IoT are in their infancy, still there are many problems and IoT use cases that are not completely realised. Even though IoT has a vast implementation in conventional routine creating scenarios with almost every network technology to extract information, bringing improvement in daily life and developing a smart ecosystem. In this section, we undergo an extensive review of the existing solution of IoTs based on SDN. Few of IoT implementation in the context of SDN based control and management is discussed below. SDN integration in current trends of IoT is a research question till yet. In this regard, many studies have been generated in the campuses and on the industrial level to get full advantage of programmability from SDN and Virtualization from NFV.

### A. SDN Based D2D communication in LTE

Long Term Evaluation (LTE) is a communication standard evolved from Third Generation Partnership Project (3GPP) known as UMTS (Universal Mobile Telecommunication System) and introduces Multi Input multi-output (MIMO) to ensure high-speed data transmission at a higher data rate of 300Mbps peak downlink and 75 Mbps peak uplink [22]. It also provides connectivity of cellular network with the Internet using IP network equipment LTE support high data required services such as Voice over IP (VoIP), Video conferencing and multimedia streaming in a cellular network. It uses multiple radio access techniques and uses both Time Division Duplex (TDD) and FDD for downlink and uplink high data rate communication and improves spectrum efficiency. The working component of LTE are User Equipment (UE), eNodeB (access point), and EPC i.e. Evolved Packet Core. UE is actually a mobile used to link the user with the access network. The access network is an Evolved UMTS Terrestrial Radio Access. A general architecture for SDN based LTE is shown in Figure 4.

LTE, a major contributor in IoT, promise high data rate and low latency but despite these facts, LTE technologies encounter many issues of centralised control, Scalability and QoS challenges in the network. Centralised management and spectrum adjustment by operator minimises the automatic and dynamic control and management of the cellular network. In this context, several studies have been conducted based on the integration of LTE with SDN. In [14], LTE network reconfiguration is proposed using SDN based on D2D communication devices and ensure Quality of Experience (QoE) which is measured on the basis of Mean Opinion Score (MOS). Liu et al. proposed an algorithm for multi-tier LTE network reconfiguration for downlink and uplink based on a D2D communication protocol in case of congestion on the nearest eNBs. The parameters used to measure performance are download speed and waiting for the delay because of congestion in the adjacent eNBs. Savarese et al. in [15] proposed a Flexible approach for the reconfiguration and resource allocation in LTE environment when acting as IoT by observing context and connects various types of monitoring terminal devices and the Internet without human interaction. They use context-aware information and geophysical location for their proposed framework architecture for heterogeneous M2M devices over LTE/4G network with SDN controller and context-Aware Application (CAA) running over M2M server identifies the failure of certain eNB and informs SDN about the status. In CellSDN [16], Erran et al. proposed a cellular architecture based on SDN in which attribute-based policies are formulated for individual user in the LTE network and gain fine grain control over the network. CellSDN also proposed for SDN application for deep packet inspection by the local cell agent running in each switch. This local agent in CellSDN can increase scalability by reducing the excessive load on the controller.

As controller offload some of the measurement task to the local agent which can perform local control operations. In

[26], M. H. Kabir proposed cluster-based SDN controller architecture for a cellular network where the cellular area is divided into clusters controlled by a cluster controller where major functionalities are provided by SDN controller. Radio access related activities are controlled by SDN controller, which reduces the complexity in the based station, and load monitoring and session controlling is done through the controller's head in the clustered area. The cluster head controllers communicate with each other via controller services.

Legacy IoT mostly using IEEE802.15.4, ZigBee or 6LoWPAN (IPv6 over low power wireless Local Personal Area Network) protocol as communication protocol but 6LoWPAN protocol does not fulfil the required bandwidth need of IoT devices and do not create an efficient routing. An architecture framework is presented in [27], which uses SDN as the management platform for 6LoWPAN devices. SDN based Management Framework for IoT Devices is proposed in [28]. The author used SDN controller and three reference point for communication between different network entities and SDN Controller and focus on the transaction between M2M.

The communication between the private network and the public network is done through Network Address Translator (NAT), which exhaust when the number of devices increases in the network due to its centralised nature. Distributed NAT Traversal using SDN is used for managing IoT traffic by distributing the load on the SDN-enabled devices/switches and in result transmission delay is reduced [29]. The legacy NAT traversal scheme has many disadvantages as increased workload on the relay server, or inflexible P2P communication as required by IoTs, and performance degradation due packet modification and processing on each packet. But this is not an efficient way as the central SDN controller may also suffer the aforementioned problems in the NAT and NAT Traversal schemes also there is a single point of failure due to a centralised server.

Due to the huge amount of data produced by IoT devices and billions or devices are connected to IoT network, flow management is not an easy task. In cased of SDN based IoT architecture, where the controller is responsible for making flow rules, their installation at the gateway incur delay and degrade the performance of the network. This flow rule installation is hype when flows are installed reactively on demand. In [30], Bull et al. proposed pre-emptive flow rule installation by monitoring and learning the periodic behaviour of IoT network. In this proposed scheme, the flow rules are installed before the arrival of flow in the network by observing the flow history i.e. by learning switch techniques.

According to Cisco report, due to the immensely increasing IoT/mobile device and connection, Global mobile data traffic reached 3.7 Exabyte per month at the end of 2015, up from 2.1 Exabyte per month at the end of 2014 [31]. With such an immensely increased volume of data and traffic, the single centralised controller is not sufficient to handle generated traffic and flow management. An SDN centralised controller suffer from processing pressure as only a limited amount of flow can be processed by a single controller such as on NOX, around 30k flow request per second are processed. For this purpose, distributed controller solutions for SDN were proposed such as Onix, Open Network Operating System (ONOS), and DevoFlow etc. In IoT, this traffic flow management is important in term of heavy data especially video and audio streaming, multimedia contents and online gaming etc. which need extra care for defining management rules and policies.

In [17], the author presented a detailed review of the integration of Information Centric Network (ICN) in SDN. The integration of ICN and SDN over IoT devices is not an easy task because the significant solution for security and management is lacking in realising Sensing as a Service (SaaS) in SDN based IoT devices. A. El-Mougy proposed cloud application management in ICN using SDN CP. This integrated 5G/LTE network in SDN can also suffer from security risk of single point failure, minimization of transmission rate due to shared spectrum. In [18], Usman et al. proposed a hierarchal architecture for sensor IoT integration into 5G/LTE network using SDN domain controller. The architecture is monitored by central controller and other domain controller interacts with this central controller, this central controller dynamically allocates resource leveraging a D2D communication.

### *B. Middleware solution based on SDN*

Different requirements for the two technologies are creating hazards for communication between IoT and SDNS. In [19], the interoperability of heterogeneous network in an IoT perspective is discussed and an architecture for communication between IoT and SDN environment is proposed using OMG Data Distributed Services model (OMG DDS) as middleware in which publisher/subscriber message are used for communication between different entities in a heterogeneous mode and provide scalability of a network. Similarly, CASSOWARY in [20], a provide a middleware architecture which helps in providing context aware communication in smart buildings using SDN based controller. CASSOWARY enables smart devices and SDN uses information to smartly handle the building HVAC system on the basis of distance and presence of activities or tenant in that environment.

In [21], Qin et al. enhanced the idea of Multi-network controller architecture for heterogeneous IoT network based on SDN controller for a multi-network environment such as network accessing Wi-Fi, WiMAX, LTE, ZigBee and another cellular network at the campus level and evaluated the performance by measuring delay, jitter and throughput. MINA is basically a middleware whose working principle is self-observing and adaptive, and manage the pervasive heterogeneous network. MINA takes advantage of SDN principle for flow matching and management. MINA follows SDN like layered architecture, which reduces the semantic gap between IoT and task definitions in a multi-network environment. The architecture is modelled using a Genetic algorithm and network calculus. Flow shares the same node resources and network is optimised for this resource sharing in this architecture.

WU et al. in [22], presents UbiFlow framework which provides the integration of the SDN and the IoT. UbiFlow proposed an efficient flow control and mobility management in urban multi-networks using SDN distributed controllers. In UbiFlow architecture, IoT network is partitioned into small network chunks/cluster in which each partition is controlled by a physically distributed SDN controller. The IoT devices in each partition may be connected to the different access point for different data requests. These distributed controllers coordinate to provide flow scheduling, mobility management, optimized access point selection in a consistent, reliable and scalable control order, and provide fault tolerance and load balancing for multi-network IoT. The per-device flow management and optimised access point selection are based on the multi-network capacity performed by the SDN controller, which partition the network using network calculus in the UbiFlow architecture. Ubiflow architecture is shown in Fig. 5

A representative summary of existing SDN based management Solutions for IoT given in survey are presented in Table 1.

### *C. SDN for wireless sensor based IoT devices*

Wireless sensor network defines intercommunication of spatially distributed sensor node which is generally used as monitoring agent in the disaster areas, health care, environmental condition, industrial monitoring and earth sensing etc. The most common contributor in the IoTs is sensor nodes. Wireless Sensor Network (WSN) is deployed in different scenarios according to specific need e.g., sensor deployment for a volcanic study to deep-sea measurement, in the disaster area to dark forests reading throughout day and night. Many research articles articulated the role of wireless sensor nodes in smart ecosystem and contribution of telecommunication. However, tremendous growth in IoT devices/sensor node, application, collection and analytics on data need intelligence services and new paradigms. Our focus in this study is the integration of IoT component with SDN, so we collect reading based on WSN in the context of SDN. Mostly sensor node topology is a mesh topology or a peer-to-peer topology; management and control in constrained environment are always a vigorous research area.

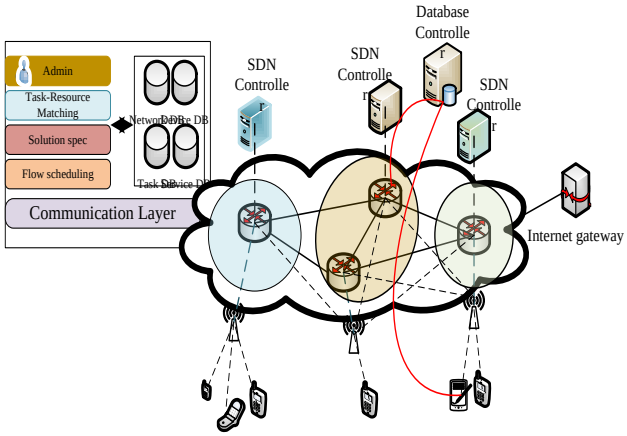


Fig. 5. UbiFlow architecture

In this context of mesh network connectivity, an interesting instigating architecture for a wireless mesh network (WMN) on the basis of OpenFlow was given by Delay et al. in [23]. In this paper, the author suggests the seamless mobility in the WMN by the use of OpenFlow. The KAUMesh test-bed allows the use of OpenFlow in WMN and provides an efficient and flexible mobility solution. In this solution, the mesh router is OpenFlow-enabled and contains multiple physical wireless cards. Multi-hop connectivity is achieved by using OLSR and data path uses local sockets to communicate with the control path component. Monitoring and Control Server (MCS) and NOX act as a controller interface and communication is done on a secure channel and handles all the flow rules. The association database contains a list of stations and the Mesh Access Point (MAPs). Connectivity graph can be obtained from gateways or may be from the QoS metrics. NOX handle routing task based on the information gathered from MCS. New rules are installed using topology database in the data plane. The associated station complies IEEE802.11standards and handover is done using IEEE802.21 standards. The OpenFlow protocol is used for setting up the flow tables, HTTP/XML for the communication between MCS and NOX and IEEE 802.21. This architecture is important as mesh connectivity play an important role in IoT scenario. However, the mobility model is suitable only for small scale while IoT implication is seen in larger context. The algorithm for the association of flow node and flow path in this architecture is undefined for MSN.

In the context of WSN management, few protocols are proposed such as SDN-WISE[24], Software-Defined Wireless Sensor Network Framework [25] and leverage SDN programmability in the WSNs. The architectural components of this approach consist of a Base Station (BS) and several sensor nodes. SDN controller operates on BS took a routing decision on the lieu of dumb sensor nodes. Sensor nodes contain flow table as in the SDN populated by controller.

WSN integration in SDN is seen in [41] with a three-layer architecture. It consists of master node/controller node, central node (OpenFlow enabled switch) and a normal node. The master node defines a routing policy for the normal node. The author et.al uses flowVisor as virtualization engine to make independent user slice in between controller and switch. Data forwarding is done by OpenFlow switch. The configuration and management are done by OpenFlow protocol, which also identifies the existing routing protocol, and work in congruence. The placement of central node is important. In this proposal, the distance is calculated based on cosine similarity formula. The central node locates in the physical centre of the cluster architecture, helps in maintaining network topology and help increasing network convergence. They name their architecture as SDWSN. Neighbouring node status is taken into account for the assigning role. Even though the author has verified their architecture through comparing its result with existing WSN protocol and find improvement in the result; the conceptual details are not very clear.

In [28], Miyazaki et al. proposed an architecture for reconfigurable WSN network on the basis of customer need

by using role injection and delivery mechanism. The role compiler generates scenarios which are injected through wireless communication. Field programmable array (FPGA) and a microcontroller unit (MCU) carry the change in the sensor nodes. The communication is done on the basis of sensor attribute. The role compiler is at the base station. The architecture consists of base stations (BSs), reconfigurable node and a server which contain role injection and delivery mechanism components. They name their architecture as SDWSN. Neighbouring node status is taken into account for an assigning role on the fly and sensor behaviour can be manipulated as per role description.

IoT devices with constraint resources are the main consideration while forming any architecture or protocol. The increased efficiency of sensor node communication is directly associated with energy management. Majority research focuses on sleep/active mode for energy restoration in the WSN. Wang et al. in [29] presented an SDN based algorithm "Energy Consumed uniformly Connected K-Neighborhood" (EC-CKN) called as SDN-ECCKN. In this architecture, a controller node calculates the overall energy of WSN. SDN-ECCKN helps in retaining energy of each node and minimises the broadcast messages from the individual node.

The multi-purpose sensor network is also addressed in [30]. Leontiadis et al. exploited NFV for sharing single infrastructure for many applications in a sensor network. They proposed a framework for multiple application scenarios on a common build infrastructure. Each node has an abstraction layer for a shared hardware which works on the overlay network and creates multiple virtual sensor networks (VNS). The bridge between application and hardware is written in TinyOS operating system. This is informally an idea of separating sensor node hardware plane for application oriented overlay VNS.

In [31], the author proposed an architecture for the integration of WSN with SDN controller. A local controller in each sensor node is responsible for MAC forwarding and some local routing decisions. A centralised controller is responsible for the long-term decision. In a sensor network, topology information collection is main challenge and different approaches are used for the information collection like packet trace, which contains detail information and Link Quality Estimation (LQE). The author suggested using lightweight LQE for collecting topology information, which can provide SDN controller with a global view of the network. This paper also proposed to take advantage of virtualization of SDN and change the object bytecode on the fly for commodity hardware. The SDN logical manipulation of virtualization and intelligent algorithms is used to get better IoT application and traffic analyser. Many of the implementation scenarios are also presented by the author.

Software Defined Wireless network (SDWN) [64], is an early effort for providing feasibility for the implementation of SDN for the wireless network. Costanzo.et al. presents architecture for Low Rate Personal Area Network (LR-PAN) management and flexible resource utilisation using SDN controller over the sink node. Sink node gathers topological information and coordinates this information to the controller, which defined rules/policies for better management. Each individual node computes RSSI factor for measuring network resource (local battery level and hop count). The rule or policies are defined by a controller implemented on a limited portion of incoming packets to safe space. However, this architecture does not support any concrete OS for SDN based IoTs and the solution for wireless infrastructure based network does not fit in the infrastructure-less plethora of WSN. A summary of SDN based solutions for sensors networks is presented in Table 2.

### *D. Software defined Radio*

The management of lower layer of the protocol stack is already introduced as Software Defined Radio (SDR) for managing the underlying complexity of hardwired implementation of the wireless network. The constituent entity of wireless communication is radio frequencies. With the increasing complexity and aggregated telecommunications technology and Radio Access Network (RAN) cross functionality is hard to obtain the desired result and need to physically intervene in radio technologies. By providing software-based radio manipulation, distinct management flexibility can uplift network performance. Constantly increasing IoT devices in billion and trillion and their communication need hardware independent implementation of network and radio connectivity.

SoftRAN [33] is proposed by Tomovic et al. which uses SDN principle in 4G LTE network. A centralised control plane abstracts the whole RAN into the geographical area. This Geographical area acts as a big base station where many radio elements i.e. physical base station are deployed under the control of the centralising controller; who manage radio resource allocation in the big base station.

The author proposed resource allocation in the form grid of three dimensions i.e. space, time, and frequency slots. The interaction between controller and radio element is done through APIs. Radioelement backup the information in the control plane. Based on this information, the controller decides to allocate resource in the domain of frequency, time and space slot. Radioelement takes some of its decision based on local information to manage the delay between controller and radioelement. Hence global network decisions are taken by controller local small resource management is done by the radio element.

SoftCell [34] incorporate SDN in the cellular core network and provide fine-grained policies for an LTE network. The contributing components in SoftCell architecture are i). Controller, ii). Access switches, iii). Core switches and iv). Middle-boxes. The controller defines policies and implement through switch level rules through middle-boxes. Traffic classification is done on the access switches. Every access switch has a local agent which caches each UE profile. In this way, local agent control of packet classification is access switch and undue burden over the controller is reduced. Controller has a global view and defined rules on the match fields i.e. policy tag, hierarchical IP address and UE identifiers.

The location and policies are embedded into packet header to avoid reclassification of the traffic. Core switches connect to the Internet through gateways fine-grained policies ensure through multi-dimensional aggregation and packet classification in asymmetric topology.

An integration of SDN and SDR in 5G network is proposed in [35] called Hybrid SDN/SDR architecture . The proposal architecture is cross layer combination of SDN and SDR for exploiting frequency spectrum and link information in 5G network. Network environment consists of spectrum and bandwidth perception in SDR layer while SDN controller can detect channel usage in the network. The cross-layer controller has used request frequency spread spectrum and is the decision maker and review flow traffic. This architecture also manages user authorization in the cross layer controller and grant access to a better band. The process of cross-layer communication between SDR and SDN starts with scanning spectrum holes.

SoftAir [36], proposed by Akyildiz et al. for the integration of SDN principals in 5G network by exploiting cloudification and network virtualization of a resilient network. The architecture provides mobility aware load balancing and resources efficient allocation through virtualization. The network architecture is based on software-defined switches and BSs which be dynamically programmed. The aggregated control is provided by NFV creating multiple virtual networks with independent protocols and resource allocation algorithms. Data plane comprises of SD-RAN and SD-core network nodes, which are OpenFlow-enabled. Data plane monitoring is done through OpenFlow and Common Public Radio Interface (CPRI). All management policies are defined at central control plane, which enables cloud orchestration. Traffic management module in control plane selects an optimal path in mobility aware context. QoS applications are carried out through distributed traffic classification module in the control plane. Overall, SoftAir presents a detailed and complete architecture of 5G cellular network management based on SDN and provide end-to-end QoS guaranty.

SDN&R [37] present a merger of SDN and SDR for IoT network and provide integrated management of diverse IoT network. SDN decouple the control plane from data plane and SDR is used to maintain radio status information in the control plane implemented on a base station (BS). The OpenFlow-enabled control plane performs radio control on the BS and cognitive edges (CE). The CE obtains the complete view of the radio spectrum. The packet processing is done on the controller connected to BS via a secure channel. The SDN-enabled cognitive radios resource management. This architecture is the detailed footprint of SDN integration in a cellular network for managing resources that are highly demanded in IoT network. A comparative review of studies literature Cellular IoT Solutions on SDN basis are presented in Table 3.

### *A. SDN based IoT Management*

In a heterogeneous network like in IoT, where diverse technologies are interplaying and exchange information. In such networks, the management becomes very complex. The configuration, reconfiguration, resource allocation and even the pattern of intercommunication becomes extremely difficult. SDN, due to its decoupled nature, separate control plane from data plane offer programmability and management from a centralised server having a global view of the network status. SDN play a vital role in the management of such heterogeneous network. M2M communicating devices are managed through leveraging SDN control plane in [28]. The proposed framework is a two-tier architecture consisting of control plane and data plane and devices are IP enabled. These devices are populated with routing table as in the SDN-enabled switches. Controller has a complete view of the network. If a breakdown observed between devices and gateways, the controller does network reconfiguration. The communication between devices is used three reference points Mx, Gx, Gnx. The device kept its information and its neighbour information in the form of a file such that any change in the file is manipulated on controller instruction.

The management of a heterogeneous smart environment is quite complicated compared to a homogeneous M2M communication. Boussard et al. [53] proposed SDN based control and management framework for IoT devices in a smart environment. In their management framework, called "Software-Defined LANs (SD-LAN)", devices are organised and grouped in the order of requesting services from the user. The framework is a four-layer architecture consisting of (i) task description (ii). Service description, and (iii). Flow scheduling and low-level communication. This framework uses Universal Plug and Play (UPnP) and Simple Service Discovery Protocol (SSDP) discovery for the incoming device in the SD-LAN network. A virtual topology is created for SD-LAN devices based on services requirement such as audio, video, online game streaming etc.

The legacy routing waste resources and uses link unfairly. In the case of packet loss, the correlated latency also increases with caused performance degradation. In wired network packet drop may be caused by congestion on the link but in large-scale IoT devices (mostly wireless ), this re-routing cause a Ping-Pong situation and the overall network performance degraded in case of any packet drop detected whether caused by a small interval. Context-aware IoT architecture

IoT applications occupy every domain of life and effect socio-economic factors such as health care, security, disaster management, remote access to things etc. In this context, D2D communication and coordination can play an important role where devices can seamlessly configure and reconfigure network without human intervention. Environment monitoring can be done if the IoT objects are implemented in a context-aware mode of communication. In [15], G. Savarese proposed a context-aware framework for LTE communication for D2D.

TABLE I. THE COMPARISON OF EXISTING SDN BASED MANAGEMENT SOLUTIONS FOR IOT

| Architecture | management | Architecture | Control/data plane decoupling | Protocol used | scalability | Simulation Tools | benefit | Limitation |
|---|---|---|---|---|---|---|---|---|
| MINA[21] | Flow scheduling and management | Redefining the controller architecture based on DDS middleware and decouple the services and actual mechanism of traffic forwarding | Centralized controller | OpenFlow like protocol and IP protocol | - | Qualnet | Better performance and flow scheduling | layered controller design is critical to the management and still not addressed |
| Publish/subscriber-SDN[19] | Services/application management and resource management | It uses modular approach and translate user services message into SDN flow using DDS at the gateway | Centralised controller on the access point | COAP and OpenFlow | High | - | Scalability, mobility and security. Efficient handover | No validation proved through experiment or simulation results |
| CASSOWARY[20] | Profile and policy management | Context-aware sensor deployment using cassowary middle box on SDN controller. Network is divided into the In-Memory data grid | Device controller smart equipment | AMQP | Medium | cloudSim/ cassowary written in JAVA | Energy efficient and security profile and authentic access | Scalability |

TABLE II. THE COMPARISON OF EXISTING WSN- SDN SOLUTIONS

| Architecture | management | Architecture | Control/data plane decoupling | Protocol used | scalability | Simulation tools | benefit | Limitation |
|---|---|---|---|---|---|---|---|---|
| SDN_WSN[26] | Topology discovery and management | Centralised controller with three reference points | M2M communication between centralised controller and node | OpenFlow | Low | - | Intercommunication between devices and sensor node using gateways and centralised controller | Undefined functionality and implementation, no proof of evaluation. |
| WSN-SDN[27] | Sensor network flow management | WSN cluster with centralised controller monitored and controlled by Master SDN controller | Centralised master controller | OpenFlow/ distance aware routing protocol | Low | MATLAB | Optimal path selection, routing strategy adjustment on the network condition | Implementation of master and central controller is not clear, No proof of validation, |
| SD-WSN[28] | Infrastructure management and reconfiguration of sensor network | FPGA | Micro-controller | COAP | Low | - | Programmable reconfiguration of network | Hardware bounded and device dependency |
| ECCKN [29] | Energy management | Dumb data plane node dynamically associate with centralised controller where energy efficient algorithm ECCKN run to calculate routing on the basis of residual energy | Centralised controller with dumb data plane | ECCKN and OpenFlow | Undefined | - | Reduced total transmission time and centralised control | SDN implementation is not clear and protocol interaction is not specified |
| Senshare [30] | Open access Infrastructure management | Decoupling between infrastructure and | Dedicated overlay controller | Collection tree protocol (CTP) | Low | - | Support for multiple sensing | SDN controller implementation is not clear on overlay network |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | application | | | | | applications reduced cost | |
| **Integrate WSDN[31]** | Management platform for using virtual machine in-network Processing (INNP) | Local controller in each sensor node which interacts with a centralised controller. INNP is done through VM in the node platform | Centralised controller and local controller | Contiki OS on each local controller | Low | Packet tracer trace the footprint of messaging and LQE | Flexible using commodity off the shelf device, reducing cost | Missing evaluation for behaviour and performance of WSN |
| **SOF [32]** | Flow management | INNP in data plane and flow-based packet forwarding | Centralised controller and distributed data plane | Sensor OpenFlow (SOF) | Low | - | handling peer compatibility, address classification, reduce setup latency, high throughput | Theoretical idea and not experimentally proved |
| **SDN-WISE[24]** | Localisation of distributed sensor in a centralised controller, energy management, | Centralised controller with dumb sensor node having flow table like OpenFlow flow table which is preinstalled with flow rules | Centralised controller, dumb data plane | OpenFlow | medium | - | The state-full approach, reduce information exchange. Mobility, reconfiguration and localisation of | Lacking security and reliability. In-depth architectural details are missing |

This paper briefly describes the LTE network, M2M communication and an integration of LTE and D2D based on SDN in term of context-aware monitoring of LTE eNodeB that is responsible for allocating radio resources and scheduling traffic according to the QoS LTE network.The collected contextual information of LTE network, in the case of link failure or change in the network, is sent to the Context Aware Application (CAA) running on the M2M server where SDN controller can react to this change, reconfigure LTE network and allocate LTE resources in a flexible fashion.

Jararweh et al. in [10] proposed a comprehensive framework model for software defined system for IoT for the management and control of IoT devices in the heterogeneous network. The main focus is on the storage and security issues created in heterogeneous IoT network. The data generated and collected in IoT environment is immensely high which create storage issues. Some solutions propose the use of virtualized/software storage like in [38], where physical storage is abstracted by software storage and build the storage control operation in the centralised controller. Jararweh et al. use this architecture into the IoT environment. The main idea of collecting data from the sensor board which is aggregated on the IoT Bridge and send to SDSec controller for security checking. They use authentication and authorization for ensuring only authorise access. Afterwards, data is sent to IoT controller for rules definition for the collected data with the help of routing and controlling policies from SDN controller. And these rules are stored in the SDStore module of the framework which is used by the different application.

Much of the work has been done for the migration of IoT from a legacy network to SDN. In this regard, much-cited paper [21] by Qin et al. who proposed IoT architecture for flow scheduling based on Multi-network Information Architecture (MINA) with layer SDN controller. (The IoT tasks are usually depicted in an abstract manner and they are independent of underlying network and device resource specifications). In this proposed architecture, the authors proposed semantic modelling for the high-level task and low-level resource specifications and represent IoT task as hierarchal semantic task and parameters are written in term of ontological concepts. The Task plans are stored in task Knowledge Base and resources with capabilities are stored in resource Knowledge Base. The IoT task is matched with task KB and submits to an analyser, which extracts both KBs, find resources with capabilities, and provide and appropriate solution, which is then mapped with the service solution specification. Information for resource mapping is obtained from Network information Base or DB. Afterwards, flow scheduling is done on the basis of state information provided by MINA state global information view. The QoS service is analysed using network calculus model and path is obtained by using Genetic Algorithm (GA) where each flow has a chromosome, which is a path between source and destination, and genes are considered as nodes on that network. The implantation is done in the Qualnet simulator by taking smart campus network topology. The performance metric used to delay, throughput and jitter for file-sharing, tele-audio, and video flow over the network and compared their GA scheduling with two existing SDN scheduling algorithms bin-packing and load balance algorithms and find that their results are consistent. In this paper, the author et.al did not found the flow entry overhead in the beginning and consider that their flow scheduling GA is stable.

However, the initial overhead is not negligible and it is assumed that the flow is proactively registered in the controller. In the case of wireless IoT device, there is a chance of change in the topology, which needs to reregister the flow, which create extra overhead and performance degrade.

In [55], Xiong et al. presented resource allocation architecture for SDN based IoT network. The average reward of the network is increased by considering long-term expected average reward per unit time and based on this reward optimal

resource allocation problem using MDP. The reward model is computed by assuming states and actions in each state. Using this reward, an optimal resource allocation policy is formulated using value iteration algorithm.

Ancuta. et al in [56] presented the concept of a management solution for dynamically instantiated services in an elastic environment. The information is exchanged between different entities consuming more energy when HTTP protocol is used for message forwarding. In this context, an extendable architecture open MTC is proposed and its implementation is prototyped which uses oneM2M and ETSIM2M protocol that run on Gvent API. They show that as soon as the new instance in M2M arrives, the information is an exchange between M2M management adaptors which informed the M2M connectivity manager who retains the policies for the M2M devices. This transport policy is announced. By this implementation, the scalability can be increased but there is a factor of delay as the number of devices increased in the network.

### *B. SDN-Based IoT Operating System/controllers*

The IoT devices, in general, are heterogeneous and use multiple technologies for intercommunication. Even though IoT uses multiple middlewares to reduce, the gap between application and IoT devices message passing, interoperability is still an issue to enhance the performance and increase the reusability of IoT network. To deal with this interoperability, network Operation System (NOS) play an important role in managing interoperability in heterogeneous systems. As sensor nodes and actuator are considered as a building block of an IoT network. These tiny device/motes are constraints of energy resources, storage capacity, and processing power, content-based routing etc.

However, the established OS for these tiny IoT components in a WSN based IoT network are not capable of handling interoperability on large scale and conversion of flow. For this reason, many OS, Such as Contiki [57], RIOT OS [58], Tiny OS [59], Lite OS [60] etc. were presented for WSN based IoT network. However, these operating systems are specific to the certain application, thus lacking flexibility and dynamism i.e. independent of platform in a system. A comparative analysis of these all OS is presented in Table. 4.

Still, there is no concrete OS for managing the integration of IoT and SDN. In this context, a little effort is put in developing OS for SDN based IoTs which in return create complexity in translating flow rules/policies for IoT devices. SDN is also in its infancy and it uses OpenFlow is used for bridging gap between SDN control plane and data plane. Few NOS are also available in market such as NOX, ONIX [62], Maestro [63], OpenDaylight [14] etc. These controllers are well operated for wired SDN but these OS are not suitable for SDN-driven IoT network. This controller or OS lack support for the characteristics of IoT devices such as fundamental energy and processing constraints, data aggregation, duty cycle etc. The initiating concept of reprogramming and re-tasking in WSN was proposed in Sensor OpenFlow (SOF) [47]. SOF is three layer architecture; application layer, a control layer and data plane layer. The application layer consists of all applications necessary for managing query applications, data processing applications etc. Control layer consisting modules are "sensor re-configuration" module and "query strategy control" module and perform flow-based forwarding in the data plane consisting FDs sensor nodes. Forwarding plane forwards the sensor flow in the order defined by the controller. However, flow creation and management was a challenge in SOF and the overhead created due to control traffic can dim the expected outcome of SOF. To overcome these limitations, complexity is added and simplicity is reduced. For the sack of providing flexibility and simplicity in WSNs through SDN, an operating system solution based on SDN was proposed by Galluccio, et al. in [38], named as SDN-WISE; an architecture and operating system for WSN support duty cycle and data aggregation and provide a state-full solution for SDN. The consisting data structures of SDN-WISE are the WISE States Array, the Accepted IDs Array, and the WISE Flow Table. The communication between sensor nodes and other controller is done through WISE-Visor resemble in the functionality of FlowVisor [65] which is switching virtualization approach in SDN. The introduced adoption layer performs translation between the sensor node and WISE-Visor and decouples data plane and control in the SDN based sensor network. SDN-WISE is a state-full approach and defines its policies on the basis of state description, shown in adopted example from [66] which depict policy implementation for a packet if its threshold or measure is less than a certain threshold ($X_{thr}$) and it is generated by node A as shown in Fig. 6. Details of the studied literature in the context of the controller and operating systems in sensor networks are given below in Table 4.

SDN-WISE ensure the minimum number of information exchange and holistic support for different protocols and node design. Christos et al. do an enhancement in SDN-WISE in [67]. The authors propose an OS based on Open Network Operating System (ONOS) [68] and integration of SDN-WISE and OpenFlow network in a seamless manner. An OpenFlow-enabled device can interact with a WSN network through ONOS.

### *C. SDN security framework for IoT*

In the most recent IoT arena, billions of Internet-connected physical objects produces the bulk of data within few milliseconds whose storage, processing, automation, and management is an intensive task. These devices are potentially under threat due to unbounded connectivity and communication over wired and wireless transmission medium due to lack of standard security protocol/architecture for IoTs. SDN is considered a powerful technology of having centralised control over the information flow in the network and provide a preemptive security policy. The IoT system becomes more vulnerable to security risks when they are monitored from a centralised controller as SDN based IoT network.

Little considerations of security aspect are witnessed in SDN based IoT network. In [39], Sahoo et al. proposed a secure architecture for IoT network based on SDN. There are five basic security properties which need to be under consideration while defining a security model. These security characteristics are Confidentiality, integrity, availability, authentication and non-repudiation [39]. Sahoo et al. proposed

their secure architecture on the basis of authentication of IoT device on the controller. In this architecture, the considered IoT is an ad hoc network in which wireless object establish a connection with the controller and controller block all the port when the connection is established and controller starts authentication. If the user is authentic, the controller starts pushing flow to that user. Few controllers in the network serve as a security guard and exchange information with each other about the user authentication. In the case of guard controller failure, some other border controller is selected as security controller.

Even though this work presents a basic layout for secure SDN based IoT network, however, the validity and correct operation are not provided.

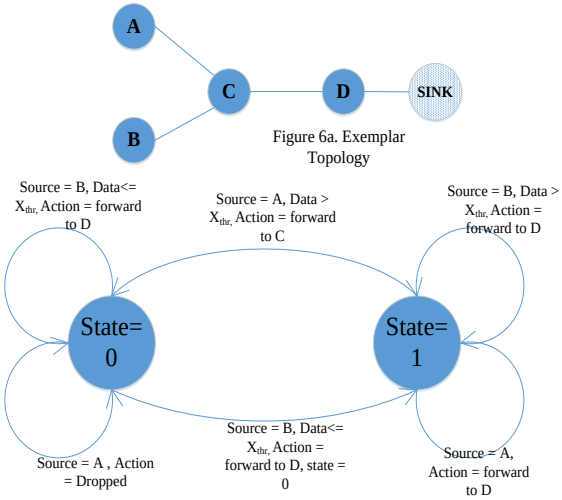


Figure 6a. Exemplar Topology

Fig. 6. Action derived from FSM in SDN-Wise

In [40], authors proposed a dynamic firewall named as Distributed Smart Firewall (DISFIRE) for secure architecture in SDN based grid network. The architecture consists of hierarchal cluster network with multiple SDN controllers. These cluster head SDN controller implement a security policy. For this purpose, they used cisco defined policy agent opFlex [41] in the controller instead of OpenFlow. The device information is exchanged between devices and any unauthorised potentially malicious device flow rule policy is deleted.

A security proposal for smart cities is presented in [42]. Chakrabarty *et al*. proposed a secure architecture based on trusted SDN controller, Black Network, Unified Registry and Key Management System in an IoT network. The security architecture ensures authentication of the heterogeneous devices. SDN controllers act as a Trusted Third Party (TTP) and provide security properties i.e. confidentiality, privacy, integrity, authentication, and routing between IoT devices. The unified registry is responsible for Identity management, availability, accounting, authentication, authorization. The shared key is used for secure communication.

Ad-hoc network in term IoT network do not provide access control and traffic monitoring in ad hoc network is not possible therefore security is a threat in ad hoc network where infrastructure is missing and connection are established reactively. In [43] Architecture is presented where each node is connected to a domain controller through an embedded virtual switch. This controller is on the edge of the network and acts as a domain controller and provide authentication of the network devices. On the authorization profile, flow entries are pushed in the access switch. Oliver *et al*. [44] proposed a SDN based IoT architecture for infrastructure and infrastructure-less network where a virtual switch is embedded in each node bounded to a controller in a domain. Devices in different domains interact with the border switch. Some of the border switches are selected as controller and these controller acts as a security controller. The security controllers provide dynamic network configuration and security policy deployment. The architecture provides Authentication of the network devices on the time of device registering with the controller.

IoT/M2M communication can leverage emergency response in case of network failure in a disaster situation and can aid the first responder in taking appropriate decisions. In [45], a security architecture for the first responder in the IoE/IoT environment is proposed using Software Defined Perimeter (SDP) protocol. Where SDP collect the IP addresses of all M2M communication capable devices and store into a logical network. When any new M2M device comes in close proximity of SDP domain, they first configure themselves in a secure SDP by using authentication credentials. SDP efficiency of authenticating secure access in the emergency response is visible, it also can data privacy and trust in the M2M communication network.

SDIoT [10] present the security of SDN based IoT network by implementing SDSec module which utilised NFV to create a virtual topology for the connected device and leverage the benefit of SDP for authentication by block all the switch port when received a request from a new flow. SDSec store information in the security database and it identifies an object by tracking authentication DB. SDN controller set flag P if everything is good otherwise flag N for negative. If the flag is set P then flow is allowed to enter and access is granted. Another security framework is proposed in [46]. In this architecture, author uses IoT agent and IoT controller that are responsible for connecting SDN controller in the SDN-enabled heterogeneous network. IoT agent is registered agent with IoT controller. SDN controller performs authentication and routing based on collected information from the IoT agents. The whole IoT network is divided into segments with its own SDN controller. Every IoT device must be connected to an OpenFlow enabled IoT device, which coordinates with segment controller. The inter-segment communication is through gateway controller. Embedded system implication in intensive health monitoring is a rich field; highly requiring security and reliability in information interchange. Cyber-attacks and malicious encroachment are very common in the Internet-connected environment and can modify the functioning of embedded systems. Security system in the embedded system does not entail high processing security techniques. Ukil *et al*. exploited the detail security threats in embedded system in [47]; proposed Secure Execution environment (SEE) mediating security model from outside security threats. Dedicated security processor

compartmentalised from non-secure mode is SEE architecture with dedicated RAM for retaining integrity and confidentiality from out the SEE code. Intrusion detection system (IDS) implementation is not easy in IoT as it requires complex mathematical computation and profile based modelling. In [48], Skowyra *et al.* exploited the idea of IDS based learning in the mobile embedded system for restraining modification from any anomaly either from inside the network or from out of the network. The OpenFlow controller contains all logic and defines rules based on state-full information. Table. 5 presents the studies literature about the security-related solution in IoT-based on SDN.

## IV. Discussion and Open Issues

The whole concept of IoT-SDN is not mature, and standardisation efforts are still under way, multiple competing alliances are trying to dominate for a global standard. We have discussed broad literature on the integration of SDN and IoT. In this study, different aspects of SDN integration in IoT technology in the context of M2M communication, LTE/IoT communication, Sensor IoT heterogeneous network are discussed. It also highlights the proposed solutions for architecture, management framework; security aspect in the SDN based IoT. A detailed overview of the observed studies is given in Table 5; which demonstrate the diversity of SDN incorporation in different IoT domains. Another important thing to notice that most of these studies are not experimentally validated; however only a representative proposal frameworks are grabbing the attention during last five years. This is because of the anticipated benefits of SDN programmability in the management of mushroom growing IoT devices. This effort could become a reference point for the researchers and developers to investigate the trending IoT application in a more controlled way; proving fast innovation and change due to technology shifts.

However, the existing solution is not fully integrated into SDN and a comprehensive architecture and framework are not established so far. Few effort are really admirable such as SDIoT, BlackSDN etc.,

where a complete framework for IoT devices is presented giving SDStorage, SDSystem and SDSec for management, security and architectural detail of IoT interplay in SDN. A major factor of lacking a comprehensive architecture for SDN based IoT is the absence of a concrete framework of IoT architecture.

SDN main characteristics lie in the wired and infrastructure-based network, while in IoT, devices are diverse in nature and different communication technologies are blended to form a heterogeneous network. This merger may be mobile in case of ad hoc network or vehicular network where dynamic allocation of resources with constraints devices need object addressing, which is still not addressed in SDN, based IoTs

Another issue in the IoT network is content addressing and context awareness in services provisioning with QoS support, which is still not addressed in any work. Existing transport protocols fail in the IoT scenarios since their connection setup and congestion control mechanisms may be useless; furthermore, they require excessive buffering to be implemented in objects. Also, the traffic pattern in IoT is different from the traditional network traffic even different from SDN/OpenFlow data flow; require excessive intention from the researchers.

In SDN, loT of control traffic consume bandwidth and hence degrade the spectral efficiency in the IoT Devices. Also, the battery power is highly vulnerable to this massive control traffic. In IoT devices, the traditional security characteristic is hard to implement, the authentication and authorization require a storing of authentication profiles in the minute storage. Well-known traditional network security cannot be applied in IoT. SDN centralised control plane may suffer from denial of services attack and man in middle attack. Due to the huge amount of data produced in IoT network, data privacy is a critical issue in the case of M2M communication in IoT network.

The controller is still not defined for the IoT. The controller took a lot of space and implemented on the server side; in that case, the instruction set produced by the SDN controller should be formatted according to the IoT devices. The single centralised controller is prone to single point failure; therefore, a need for distributed controller is a research question in IoT communication network.

## V. Qualitative Predictions for 2020

The IoT will help in establishing smart ecosystems such as smart home, smart building, smart health care unit, disaster management, smart industrialisation, nifty transportation and smart grid station etc. and eventually bring a social and industrial revolution. According to a statistic data obtained from [2], around 14.4 billion connected devices were there in 2014 and will reach up to 50 billion connected devices in 2020. The increasing trend in the IoT connected device with respect to the world population is shown in Fig. 7

IoT adopting is like a wildfire spreading across dry grass and millions of IoT-enabled smart devices are in operation like sensors, actuators, RFIDs, vehicles, PDAs, smartphone, cellular devices, wearable's, smart bulbs, smart turbines, smart arms and much more. This widespread adoption of smart object and interconnectivity has changed the market and research interest. According to a report by Gartner, Inc., around 6.4 billion devices are in play till 2016 which is 30% more than in 2015 and there is approximately 5.5 million new devices are connecting to the Internet per day. This count is immense increased and will reach to around 20.8 billion in 2020 (according to Gartner report) and will reach up to 50 billion connected devices in 2020 creating revenue of $14.4 trillion. Due to this high-expected statics, companies are bullishly spending a huge amount on IoT integration; around $656 billion were spending in 2014, which estimate a rise up to $1.7 trillion in 2020. It is estimated that there will be a 90% rise in the installation of intelligence and smart connectivity in cars until 2020, which was only 2% in 2012. This swift switch is forcing manufacturers and industries to look into broader sense and hence research trends are changes as shown in Fig. 8.According to International data corporation, around $8 billion will be generated which was only $960 million dollars in 2014; 90% compound growth rate. According to Gartner,

SDN application and infrastructure is top 10 strategic during 2015. The annual data growth rate also crosses limits in zeta-bytes in 2016 and predicted to cross up to 2.3 ZB by 2020. According to IDC, overall enterprise network revenue will grow 3.5% to reach $41.1 billion When the growth rate comes in term of SDN then according to Gartner report there is 87% increase in production in the data centre using SDN and revenue generated was $960Million in 2014and will raise to $8Billion by 2018 i.e. 734% a total rise. The increase in both domains clearly predicts a merger of two technologies and increase in the SDN based IoT production.

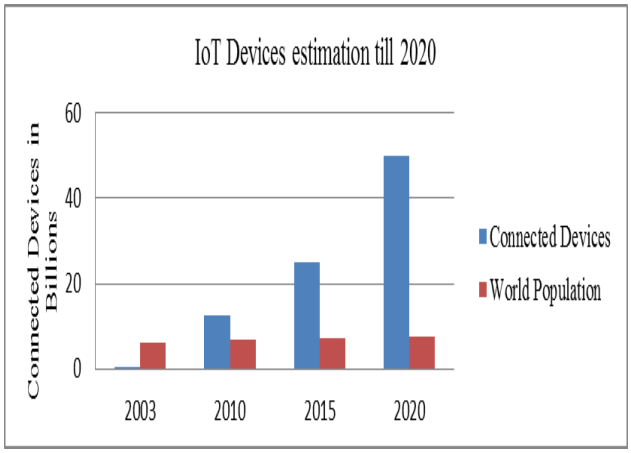


Fig. 7. Worldwide IoT connected devices

TABLE III. THE COMPARISON OF EXISTING CELLULAR IOT SDN SOLUTIONS

| Architecture | Target network | Resource management | Interface API | Purpose | Cloudification/virtualization | Control/data plane decoupling | Traffic engineering | Scalability | Simulation tools | Benefit | Limitation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **SDR** | Wi-Fi ,WIMAX | Spectrum management at software level | Smart antenna API | Increasing the spectrum allocation efficiency and virtual network/slices to support multiple wireless protocol instances | No virtualization | Centralized controller | uplink and downlink spectral efficiency | Low | MATLAB, Simulink | Independent of physical spectrum allocation | Unpredicted user behaviour, inflexible traffic engineering, Advanced spectrum management |
| **SoftRAN[33]** | 5G/ LTE | Resource management, mobility support, traffic offloading | Controller API/Femto API | To overcome the tightly bounded co-ordination in resource management | Big base station | Centralised controller and local agent at eNBs , Abstraction through slicing forming big base station | Load balancing Interference management/ SD Radio access network | Low | LTE-SIM | Radio resource management , mobility support, Traffic offloading, Reduced delay | No concrete solution, virtualization is not clear, centralised control plane and interaction between core network and RAN is not defined |
| **Hybrid SDN-SDR[35]** | 5G | Spectrum management | - | For the management of spectrum allocation and network management (e.g., bandwidth) in a 5G network | No virtualization | Centralized controller | Spectrum resource management and network resource management | low | MatLab | Power saving and optimisation | Cross-layer controller, security |
| **SoftCell [34]** | Cellular network | Fine grain policies management. | Open Flow API | Modification in the core network | Minimum virtualization | Logically centralised controller, local | MPLS and slandered routing as in OpenFlow | high | SoftCell implemented on Floodlight | Dynamic traffic offloading, efficient routing, | Fine grain service policies |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | agent SD-RAN (BS) | | | controller and micro-benchmarking using bench | minimising the state in the core network | |
| **SoftAir[36]** | 5G | Distributed traffic classification, fine grain virtualization, network management (routing) | Open Flow & CPRI | network function cloudification and network virtualization | Fine grain virtualization | SD-BS, SD-switch, BS-clustering | Collaborative processing, scheduling and mobility management | high | - | Flexible platform for fully & partially centralised architecture | Security issue not addressed |
| **cellSDN [16]** | Cellular network | Mobility management and policy control management | NOS | virtualization | Basic support for virtualization | Centralised control plane, local control agent ate BS | MPLS traffic labelling or VLAN tags | Low | - | Seamless mobility management and fine grain control due Local agent | No proof of concept and evaluation of the proposed scheme, vague traffic engineering handling using MPLS/VLAN tags |

TABLE IV. WSN BASED OS IN IOTS

| Operating System | Action | Programming Language | RAM required (Kb) | Kernel Implementation | Service management | Kernel management | Model |
|---|---|---|---|---|---|---|---|
| **Contiki** | Event based | C | 2 | Preemptive multithreading | Dynamic | Hybrid | - |
| **RIOT OS** | Task based | C/C++ | 1.5 | Multithreading | Dynamic | Static | - |
| **TinyOS** | Event | NesC | 1 | Partial | Static | Dynamic | Concurrency model |
| **Lite OS** | Event based | C | 4 | Multithreading | Dynamic | Dynamic | hierarchical file system |
| **SDN-WISE** | Event based | Java | 10 | State-full | State-full | Dynamic | Modular |
| **ONOS** | Event based | Java | 8 | - | - | - | modular |

TABLE V. SDN-BASED IOT SECURITY SOLUTIONS

| Approach | Security parameter | Network | description | Limitations |
|---|---|---|---|---|
| secured SDN framework **[39]** | Authentication | Ad hoc network | SDN controller block all switch port on receiving new flow and start authentication | Not prove implementation or simulation, only a theoretical framework |
| DISFIRE**[40]** | Authentication & authorization | Grid network | hierarchal cluster network with multiple SDN controllers implement a dynamic firewall to ensure authorization | Evaluation of framework lacking. The protocol used is opflex which is not practically tested |
| Black SDN**[42]** | Location Security, Confidentiality, Integrity, Authentication And Privacy. | Generic IoT/M2M communication | secure the meta-data and the payload by encryption in the link layer and use SDN controller as TTP | Scalability in black network will create hazard in providing complete security |
| SDP**[45]** | Authentication | Ad hoc network/M2M communication | SDP collect the IP addresses of all M2M communication capable devices and store into a logical network. And authenticate on the basis of information stored | Scalability will encounter performance in case of IoE |
| SDIoT**[10]** | Authentication | Generic IoT network | It utilised SDSecurity mechanism leveraging NFV and SDP for ensuring secure access in the network by authentication. | Hard to manage the large network in case of single SDSec logical element. An experimental evaluation is lacking |
| **[43][44][46]** | Authentication, security policy at security controller | Generic IoT | Domain controller and edge controller for security and intercommunication between | Lacking proof for concept, not tested not evaluated |

| | | | | |
|---|---|---|---|---|
| | | | different domain/segments | |
| SEE [47] | Confidentiality, Integrity | Embedded devices/System | Theoretical concept of encountered security threats in an embedded system | Processing slows down |
| L-IDS [48] | Learning network IDS | Mobile embedded devices (MEB) for the institutional site. | Mobile embedded devices dynamically form connection with the infrastructure where the possible attacker can attack MEB and | A Large number of control message interchange creates congestion on the controller. Experimental validation in not done yet. |

TABLE VI. DESCRIPTIVE SUMMARY OF IMPORTANT SDN-IOT SOLUTION FRAMEWORKS

| Approach | purpose | Implementation domain | Year | Operating system/controller |
|---|---|---|---|---|
| SDN-6LoWPAN [49] | NFV for bandwidth utilization | IPv6 local WPAN | 2015 | Centralized SDN controller |
| SDN-M2M [50] | Network configuration and resource management | M2M communication devices | 2014 | Centralised SDN controller |
| MINA[21] | Flow scheduling and management | Middleware | 2014 | Centralized controller |
| Publish/subscribe-SDN[19] | Services/application management and resource management | Generic IoT | 2015 | Centralized controller |
| CASSOWARY[33] | Profile and policy management | WSN | 2015 | Centralized |
| SDN_WSN[46] | Centralized controller with three reference points | WSN | 2014 | Centralized controller |
| WSN-SDN[41] | Sensor network flow management | WSN | 2014 | Hierarchal controller (cluster and master controllers) |
| SD-WSN[42] | Infrastructure management and reconfiguration of sensor network | WSN | 2014 | FPGA microcontroller |
| ECCKN [29] | Energy management in sensor network | WSN | 2016 | Centralised controller with dumb data plane |
| Senshare [44] | Open access Infrastructure management | Sensor networks | 2012 | Dedicated overlay controller |
| Integrated WSDN-[45] | Management platform for using virtual machine in-network Processing (INNP) | WSN | 2015 | Local and centralised controller |
| SOF [47] | Flow management | WSN | 2012 | Centralised controller and distributed data plane |
| SDN-WISE[38] | Localisation of distributed sensor in a centralised controller, energy management | WSN | 2015 | Centralized controller |
| SDR | Spectrum management at software level | Wi-Fi ,WIMAX | 2012 | Centralised control plane |
| CellSDN[16] | | Cellular network | 2012 | |
| SoftRAN[33] | Resource management, mobility support, traffic offloading | 5G/LTE | 2013 | Big base station |
| SoftCell[49] | Fine grain policies management. | Cellular network | 2013 | Logical centralized controller |
| Hybrid SDN-SDR[35] | Spectrum management | 5G | 2014 | Centralized controller |
| SoftAir[36] | network function cloudification and network virtualization | 5G | 2015 | SD-Centralized controller |
| secured SDN framework [39] | Authentication | Ad-hoc networks | 2015 | SDN controller block |
| SDP[45] | Authentication | Ad hoc network/M2M communication | 2015 | Central controller and local agents |
| DISFIRE[40] | Authentication & authorization | Smart Grid network | 2016 | hierarchal cluster network with multiple SDN controllers |
| Black SDN[42] | Location Security, Confidentiality, Integrity, Authentication And Privacy. | Generic IoT/M2M communication | 2016 | Centralized controller |
| SDIoT[10] | Authentication & authorization | Generic IoT | 2015 | SDSec module on SDN controller |
| SEE [47] | Confidentiality, Integrity | Embedded system | 2011 | - |
| L-IDS [48] | Learning network IDS | Embedded system | 2013 | OpenFlow controller |

## VI. CONCLUSION

IoT is a new norm of connectivity, enabling smart ecosystem. It is changing the way we think to communicate with an object in out surroundings and improving the quality of life. However, IoT lacks programmability, agility, security and data management due to the huge amount of data produced. To meet the need of customer requirement, it is highly anticipated use programmability and centralised control for IoT management. In SDN, control plane and data plane are decoupled, which hide the high-level implementation of the low-level forwarding devices. In this paper, we have surveyed the existing solution for the integration of SDN control plane in IoT network. In this work, first, we have discussed the existing for the IoT management based on SDN centralised control plane in different IoT contributors, summarising architectural details and its evolution, and then outline the unresolved issues in this merger and reported some predictions for the world in 2020.

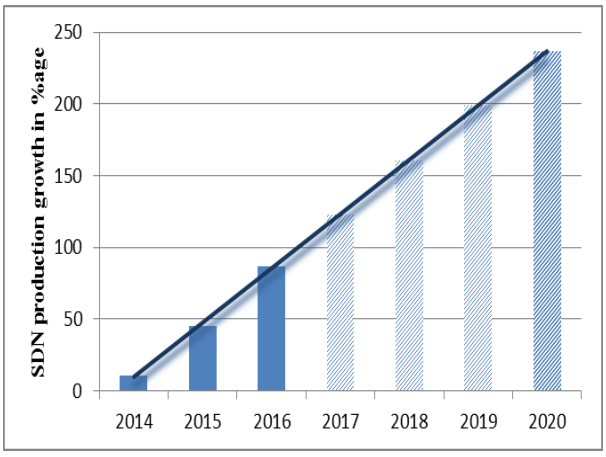


Fig. 8. SDN growth in data centers prediction for 2020

## REFERENCES

[1] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," Comput. Commun., vol. 54, pp. 1 – 31, 2014.

[2] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proc. IEEE, vol. 103, no. 1, pp. 14–76, 2015.

[3] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE Commun. Surv. Tutorials, vol. 16, no. 3, pp. 1617–1634, 2014.

[4] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," IEEE Commun. Surv. Tutorials, vol. 16, no. 4, pp. 1955–1980, 2014.

[5] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: A survey," IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 493–512, 2014.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Commun. Surv. Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," IEEE Trans. Ind. Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[8] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A survey on Internet of things from the industrial market perspective," IEEE Access, vol. 2, pp. 1660–1679, 2014.

[9] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu, "Study and application on the architecture and key technologies for IOT," in Multimedia Technology (ICMT), 2011 International Conference on, 2011, pp. 747–751.

[10] N. Bizanis and F. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," IEEE Access.

[11] T. D. N. Gray Ken, SDN: Software Defined Networks.

[12] N. Gude, ., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., & Shenker, S., "NOX: Towards an Operating System for Networks," SIGCOMM Comput Commun Rev, vol. 38, no. 3, pp. 105–110, Jul. 2008.

[13] W. Braun and M. Menth, "Software-Defined Networking using OpenFlow: Protocols, applications and architectural design choices," Future Internet, vol. 6, no. 2, pp. 302–336, 2014.

[14] [14] J. Medved, R. Varga, A. Tkacik, and K. Gray, "Open daylight: Towards a model-driven sdn controller architecture," in Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, 2014.

[15] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," in Proceedings of the 9th central & eastern European software engineering conference in Russia, 2013, p. 1.

[16] N. McKeown *et al*., "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, 2008.

[17] A. Doria, Salim, J. H., Haas, R., Khosravi, H., Wang, W., Dong, L., and Halpern, J. "Forwarding and control element separation (ForCES) protocol specification," 2010.

[18] H. Song, "Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 127–132.

[19] Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. Vouk, and A. Rindos, "SDIoT: a software defined based Internet of things framework," J. Ambient Intell. Humaniz. Comput., vol. 6, no. 4, pp. 453–461, 2015.

[20] D. Evans, "The Internet of things," Evol. The Internet Is Chang. Everything Whitepaper Cisco Internet Bus. Solutions Group IBSG, vol. 1, pp. 1–12, 2011.

[21] "Internet of Things - Architecture — IOT-A: Internet of Things Architecture."

[22] "LTE Overview," www.tutorialspoint.com. [Online]. Available: https://www.tutorialspoint.com/lte/lte_overview.htm.

[23] J. Liu, S. Zhang, N. Kato, H. Ujikawa, and K. Suzuki, "Device-to-device communications for enhancing the quality of experience in software defined multi-tier LTE-A networks," IEEE Netw., vol. 29, no. 4, pp. 46–52, 2015.

[24] G. Savarese, M. Vaser, and M. Ruggieri, "A Software Defined Networking-based context-aware framework combining 4G cellular networks with M2M," in Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on, 2013, pp. 1–6.

[25] L. Erran, L. Z. Morley, and M. J. Rexford, "Cellsdn: software-defined cellular networks," 2012.

[26] M. H. Kabir, "A Novel Architecture for SDN-based Cellular Network," Int. J. Wirel. Mob. Networks, vol. 6, no. 6, p. 71, 2014.

[27] M. M. Mazhar, M. A. Jamil, A. Mazhar, A. Ellahi, M. S. Jamil, and T. Mahmood, "Conceptualization of Software Defined Network layers over Internet of things for future smart cities applications," on Wireless for Space and Extreme Environments (WiSEE), 2015 IEEE International Conference on, 2015, pp. 1–4.

[28] H. Huang, J. Zhu, and L. Zhang, "An SDN_based management framework for IoT devices," in Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and

Communications Technologies (ISSC 2014/CIICT 2014). 25th IET, 2014, pp. 175–179.

[29] G. Kim, J. Kim, and S. Lee, "An SDN based fully distributed NAT traversal scheme for IoT global connectivity," in Information and Communication Technology Convergence (ICTC), 2015 International Conference on, 2015, pp. 807–809.

[30] P. Bull, R. Austin, and M. Sharma, "Pre-emptive Flow Installation for Internet of Things Devices within Software Defined Networks," in Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, 2015, pp. 124–130.

[31] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper," Cisco.

[32] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," IEEE Commun. Mag., vol. 53, no. 9, pp. 48–54, Sep. 2015.

[33] P. Kathiravelu, L. Sharifi, and L. Veiga, "Cassowary: Middleware Platform for Context-Aware Smart Buildings with Software-Defined Sensor Networks," in Proceedings of the 2Nd Workshop on Middleware for Context-Aware Applications in the IoT, New York, NY, USA, 2015, pp. 1–6.

[34] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A Software Defined Networking architecture for the Internet-of-Things," in 2014 IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1–9.

[35] D. Wu, D. I. Arkhipov, E. Asmare, Z. Qin, and J. A. McCann, "UbiFlow: Mobility management in urban-scale software defined IoT," in 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 208–216.

[36] A. El-Mougy, M. Ibnkahla, and L. Hegazy, "Software-defined wireless network architectures for the Internet-of-Things," in Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th, 2015, pp. 804–811.

[37] M. Usman, A. A. Gebremariam, U. Raza, and F. Granelli, "A Software-Defined Device-to-Device Communication Architecture for Public Safety Applications in 5G Networks," IEEE Access, vol. 3, pp. 1649–1654, 2015.

[38] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless Sensor networks," in 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 513–521.

[39] A. D. Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in Communications (QBSC), 2014 27th Biennial Symposium on, 2014, pp. 71–75.

[40] P. Dely, A. Kassler, and N. Bayer, "OpenFlow for wireless mesh networks," in Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on, 2011, pp. 1–6.

[41] Z. Han and W. Ren, "A novel Wireless Sensor Networks structure based on the SDN," Int. J. Distrib. Sens. Networks, vol. 2014, 2014.

[42] T. Miyazaki S. Yamaguchi, K. Kobayashi, J. Kitamichi, S. Guo, T. Tsukahara, and T. Hayashi,, "A software defined wireless sensor network," in Computing, Networking and Communications (ICNC), 2014 International Conference on, 2014, pp. 847–852.

[43] Y. Wang, H. Chen, X. Wu, and L. Shu, "An energy-efficient SDN based sleep scheduling algorithm for WSNs," J. Netw. Comput. Appl., vol. 59, pp. 39–45, 2016.

[44] I. Leontiadis, C. Efstratiou, C. Mascolo, and J. Crowcroft, "SenShare: transforming sensor networks into multi-application sensing infrastructures," in European Conference on Wireless Sensor Networks, 2012, pp. 65–81.

[45] M. Jacobsson and C. Orfanidis, "Using software-defined networking principles for wireless sensor networks," in 11th Swedish National Computer Networking Workshop (SNCNW), May 28-29, 2015, Karlstad, Sweden, 2015.

[46] H. Huang, J. Zhu, and L. Zhang, "An SDN_based management framework for IoT devices," in 25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), 2014, pp. 175–179.

[47] T. Luo, H.-P. Tan, and T. Q. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," IEEE Commun. Lett., vol. 16, no. 11, pp. 1896–1899, 2012.

[48] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013, pp. 25–30.

[49] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, New York, NY, USA, 2013, pp. 163–174.

[50] H. H. Cho, C. F. Lai, T. K. Shih, and H. C. Chao, "Integration of SDR and SDN for 5G," IEEE Access, vol. 2, pp. 1196–1204, 2014.

[51] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: A software-defined networking architecture for 5G wireless systems," Comput. Networks, vol. 85, pp. 1–18, 2015.

[52] S. Namal, I. Ahmad, S. Saud, M. Jokinen, and A. Gurtov, "Implementation of OpenFlow-based cognitive radio network architecture: SDN&R," Wirel. Networks, vol. 22, no. 2, pp. 663–677, 2016.

[53] M. Boussard, D. T. Bui, L. Ciavaglia, R. Douville, M. Le Pallec, N. Le Sauze, and F. Santoro, "Software-Defined LANs for Interconnected Smart Environment," in Teletraffic Congress (ITC 27), 2015 27th International, 2015, pp. 219–227.

[54] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, and A. Rindos, "SDStorage: A Software Defined Storage Experimental Framework," in Cloud Engineering (IC2E), 2015 IEEE International Conference on, 2015, pp. 341–346.

[55] X. Xiong, L. Hou, K. Zheng, W. Xiang, M. S. Hossain, and S. M. M. Rahman, "SMDP-Based Radio Resource Allocation Scheme in Software-Defined Internet of Things Networks," IEEE Sensors J., vol. PP, no. 99, pp. 1–1, 2016.

[56] A. A. Corici, R. Shrestha, G. Carella, A. Elmangoush, R. Steinke, and T. Magedanz, "A solution for provisioning reliable M2M infrastructures using SDN and device management," in Information and Communication Technology (ICoICT ), 2015 3rd International Conference on, 2015, pp. 81–86.

[57] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," in Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Washington, DC, USA, 2004, pp. 455–462.

[58] E. Baccelli, O. Hahm, M. Gunes, M. Wahlisch, and T. C. Schmidt, "RIOT OS: Towards an OS for the Internet of Things," in Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, 2013, pp. 79–80.

[59] P. Levis, Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., and Culler, D., "TinyOS: An Operating System for Sensor Networks," in Ambient Intelligence, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 115–148.

[60] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, "The LiteOS Operating System: Towards Unix-Like Abstractions for Wireless Sensor Networks," in Proceedings of the 7th International Conference on Information Processing in Sensor Networks, Washington, DC, USA, 2008, pp. 233–244.

[61] T. Koponen, Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M.and Shenker, S. "Onix: A Distributed Control Platform for Large-scale Production Networks," in In Proc. OSDI, 2010.

[62] E. Ng, "Maestro: A system for scalable OpenFlow control," Rice Univ., 2010.

[63] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software Defined Wireless Networks: Unbridling SDNs," in 2012 European Workshop on Software Defined Networking, 2012, pp. 1–6.

[64] R. Sherwood, G. Gibb, K. K. Yap, G. Appenzeller, M. Casado, N. McKeown, and Parulkar, G "Flowvisor: A network virtualization layer," OpenFlow Switch Consort. Tech Rep, pp. 1–13, 2009.

[65] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless Sensor networks," in 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 513–521.

[66] A. C. G. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Towards a software-defined Network Operating System for the IoT," in Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, 2015, pp. 579–584.

[67] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, and G. Parulkar, "ONOS: towards an open, distributed SDN OS," in Proceedings of the third workshop on Hot topics in software defined networking, 2014, pp. 1–6.

[68] K. S. Sahoo, B. Sahoo, and A. Panda, "A secure SDN framework for IoT," in 2015 International Conference on Man and Machine Interfacing (MAMI), 2015, pp. 1–4.

[69] C. Gonzalez, S. M. Charfadine, O. Flauzac, and F. Nolot, "SDN-based security framework for the IoT in distributed grid," in 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), 2016, pp. 1–5.

[70] "OpFlex: An Open Policy Protocol White Paper," Cisco.

[71] S. Chakrabarty and D. W. Engels, "A secure IoT architecture for Smart Cities," in 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2016, pp. 812–813.

[72] O. Flauzac, C. González, A. Hachani, and F. Nolot, "SDN Based Architecture for IoT and Improvement of the Security," in Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on, 2015, pp. 688–693.

[73] F. Olivier, G. Carlos, and N. Florent, "New Security Architecture for IoT Network," Procedia Comput. Sci., vol. 52, pp. 1028–1033, 2015.

[74] R. E. Balfour, "Building the Internet of Everything (IoE) for first responders," in Systems, Applications and Technology Conference (LISAT), 2015 IEEE Long Island, 2015, pp. 1–6.

[75] C. Vandana, "Security improvement in IoT based on Software Defined Networking (SDN)."

[76] A. Ukil, J. Sen, and S. Koilakonda, "Embedded security for Internet of Things," in Emerging Trends and Applications in Computer Science (NCETACS), 2011 2nd National Conference on, 2011, pp. 1–6.

[77] R. Skowyra, S. Bahargam, and A. Bestavros, "Software-defined ids for securing embedded mobile devices," in High-Performance Extreme Computing Conference (HPEC), 2013 IEEE, 2013, pp. 1–7