Software-Defined Networking Challenges and Research Opportunities for Future Interest

Santhosh Katragadda Katragadda and Oluwabukola Hallel

August 31, 2024

TOPIC: SOFTWARE-DEFINED NETWORKING CHALLENGES AND RESEARCH OPPORTUNITIES FOR FUTURE INTEREST

Santhosh Katragadda, Oluwabukola Hallel

ABSTRACT

SDN is an emerging paradigm currently evidenced as a new driving force in the general area of computer networks. Many investigations have been carried out in the last few years about the benefits and drawbacks in adopting SDN. However, there are few discussions on how to manage networks based on this new paradigm. This article contributes to this discussion by identifying some of the main management requirements of SDN. Moreover, we describe current proposals and highlight major challenges that need to be addressed to allow wide adoption of the paradigm and related technology. Plug-and-play information technology (IT) infrastructure has been expanding very rapidly in recent years. With the advent of cloud computing, many ecosystem and business paradigms are encountering potential changes and may be able to eliminate their IT infrastructure maintenance processes. Real-time performance and high availability requirements have induced telecom networks to adopt the new concepts of the cloud model: software-defined networking (SDN) and network function virtualization (NFV). NFV introduces and deploys new network functions in an open and standardized IT environment, while SDN aims to transform the way networks function. SDN and NFV are complementary technologies. They do not depend on each other. Thereafter, we present existing SDN-related taxonomies and propose a taxonomy that classifies the reviewed research works and brings relevant research directions into focus. We dedicate the second part of this paper to studying and comparing the current SDN-related research initiatives and describe the main issues that may arise due to the adoption of SDN. Furthermore, we review several domains where the use of SDN shows promising results. We also summarize some foreseeable future research challenges.

Index Terms—Software-defined networking, OpenFlow, programmable networks, controller, management, virtualization, flow.

INTRODUCTION

Software-Defined Networking (SDN) is a network paradigm usually characterized by three fundamental aspects:

- A clear separation of network forwarding and control planes.
- The abstraction of the network logic from hardware implementation into software.
- The presence of a network controller that coordinates the forwarding decisions of network devices.

Given that software, in SDN, can be more easily coded, deployed, and executed, SDN turns out to be a very disruptive technology that better promotes network innovation. The need for a new network architecture The capacity of the current Internet is rapidly becoming insufficient to cater to the large volumes of traffic patterns delivered by the new services and modalities (e.g., mobile devices and content, server virtualization, cloud services, big data), which is generated due to a large number of users, sensors and applications. Existing networks built with multiple tiers of static Ethernet switches arranged in a tree structure are ill-suited for the dynamic computing and storage needs of today’s and future enterprise hyper-scale data centers, campuses, and carrier environments. This makes the introduction of any new network device or service a tedious job because it requires reconfiguration of each of the numerous network nodes. Legacy networks have become difficult to automate. Networks today depend on IP addresses to identify and locate servers and applications. This approach works fine for static networks where each physical device is recognizable by an IP address, but is extremely laborious for large virtual networks. Managing such complex environments using traditional networks is time-consuming and expensive, especially in the case of virtual machine (VM) migration and network configuration. To simplify the task of managing large virtualized networks, administrators must resolve the physical infrastructure concerns that increase management complexity. In addition, most modern-day vendors use control-plane software to optimize data flow to achieve high performance and competitive advantage. This switch-based control plane paradigm gives network administrators very little opportunity to increase data-flow efficiency across the network as a whole. The rigid structure of legacy networks prohibits programmability to meet the variety of client requirements, sometimes forcing vendors into deploying complex and fragile programmable management systems. In addition, vast teams of network administrators are employed to make thousands of changes manually to network components. The demand for services and network usage is growing rapidly. Although growth drivers such as video traffic, big data, and mobile usage augment revenues, they pose significant challenges for network operators. Mobile and Telco operators are encountering spectrum congestion, the shift to internet protocol (IP), and increased mobile users. Concurrently, data-center operators are facing tremendous growth in the number of servers and virtual machines, increasing server-to-server communication traffic. In order to tackle these challenges, operators require a network that is efficient, flexible, agile, and scalable.

DEFINITION AND EVOLUTION OF SDN

Software-defined networking (SDN) is a software-controlled approach to networking architecture driven by application programming interfaces (APIs). SDN leverages a centralized platform to communicate with IT infrastructure and direct network traffic.

SDN creates and operates a series of virtual overlay networks that work in conjunction with a physical underlay network through software. SDNs can deliver application environments as code and minimize the hands-on time needed for managing the network.

The key difference between SDN and traditional networking is infrastructure: SDN is software-based, while traditional networking is hardware-based. Because the control plane is software-based, SDN is much more flexible than traditional networking. It allows administrators to control the network, change configuration settings, provision resources, and increase network capacity—all from a centralized user interface, without adding more hardware.

There are also security differences between SDN and traditional networking. With greater visibility and the ability to define secure pathways, SDN offers better security in many ways. However, because software-defined networks use a centralized controller, securing the controller is crucial to maintaining a secure network, and this single point of failure represents a potential vulnerability of SDN.

CORE PRINCIPLES AND ARCHITECTURE OF SDN

The Control Plane: Orchestrating the Cloud

The control plane is the brain of the cloud infrastructure. It is responsible for the management, orchestration, and control of the entire cloud environment. This includes configuration, monitoring, provisioning, and lifecycle management of resources.

Key Functions

1. Resource Management: The control plane allocates and deallocates resources as needed, ensuring optimal use of underlying hardware. This includes CPU, memory, storage, and network resources.

2. Orchestration: Tools like Kubernetes use the control plane to manage the deployment, scaling, and operation of containerized applications. The control plane decides where containers should run and ensures they are distributed efficiently across the available nodes.

3. Configuration Management: The control plane maintains the desired state of the system. Tools like Ansible, Puppet, and Chef are used to automate the deployment and configuration of applications, ensuring consistency and compliance with defined policies.

Examples of Control Plane Components

- **Kubernetes:** The control plane includes the API server, etcd (for configuration storage), the controller manager, and the scheduler.

- **AWS Management Console:** Provides a web-based interface for managing AWS services, including EC2 instances, S3 storage, and RDS databases.

The Data Plane: Processing and Handling Data

Definition and Role

The data plane is where the actual data processing occurs. It is responsible for executing the workloads, handling data storage, and managing network traffic. The data plane operates under the directives issued by the control plane but performs the hands-on work of processing and transmitting data.

Key Functions

1. Data Processing: The data plane is responsible for executing the applications and processing the data. This includes running virtual machines (VMs), containers, and serverless functions.

2. Storage Management: The data plane handles data storage and retrieval. This involves managing databases, file storage systems, and block storage, ensuring data is stored reliably and can be accessed efficiently.

3. Networking: The data plane manages network traffic, including routing, switching, and load balancing. It ensures data packets are transmitted correctly and efficiently across the network.

Examples of Data Plane Components

• **Kubernetes Worker Nodes:** Run the actual containers, handling the application workloads as directed by the control plane.

• **Amazon S3:** Manages object storage, storing and retrieving data as required by applications.

• **Google Cloud VPC:** Manages network traffic, ensuring data is routed correctly within Google Cloud's infrastructure.

Benefits of Separating the Control Plane and Data Plane

1. Scalability

By separating the control plane from the data plane, cloud providers can scale each independently. The control plane can manage a large number of data plane resources, scaling up or down as needed to meet demand. This allows cloud providers to efficiently handle varying workloads and ensure high availability.

2. Resilience

Separating these planes enhances system resilience. If a failure occurs in the data plane, the control plane can detect it and take corrective action, such as restarting failed services or reallocating resources. This isolation ensures that failures in the data plane do not affect the overall management and orchestration capabilities of the control plane.

3. Flexibility

With a clear separation, development and operations teams can focus on their respective areas. Developers can design applications to run efficiently on the data plane, while operations teams can manage and optimize the control plane. This division of responsibilities enhances agility and allows for faster deployment and scaling of applications.

4. Security

The control plane enforces security policies across the data plane. This centralized management of security ensures consistent enforcement of policies, reducing the risk of misconfigurations and security breaches. Additionally, the data plane can implement isolation mechanisms to protect data and processes from unauthorized access.

IMPACT ON NETWORK MANAGEMENT AND OPERATION

Software Defined Networking (SDN) hence improves network administration, scaling and versatility by offering a more flexible manner of managing networks. Here’s how SDN contributes to each of these aspects:

- **Improved Network Management**

Centralized Control: SDN provides a separation of the control plane from the data plane in which the network is orchestrated through a controller. Obviously, this approach makes it easy to configure and externally monitor the network because all points are controlled from a single source or central address point.

Automation and Programmability: In SDN characteristic, the policies and settings of the networks are programmable hence avoiding interventions. This automation results in quick provision of network services and deploys service with minimal or no human interferences.

- **Enhanced Scalability**

Flexible Network Expansion: SDN enables the scale of resources at the network level to be easier and quicker as compared to traditional networks. Since the flow of control is at a centralized point, an increase in the number of devices or the expansion of the capacity of the network does not demand a change in the flow of control on most devices. Noticeably, the network may also have the ability to reconfigure itself with relative ease when the size and configuration of the network change.

Virtualization: Software-defined networking supports the concept of network virtualization whose function facilitates the establishment of several virtual networks on a common physical substrate. This capability enables organizations to expand their networks along various requirements without the need for extra equipment.

- **Increased Agility**

Rapid Deployment of Services: Through King SDN the application of new applications, services, and network functions are easily implemented in the system. It is easy to make modifications in real-time, to match the business requirements or the status of the network.

Adaptability to Network Conditions: SDN can adapt to network traffic patterns on a real-time basis, something like congestion or failure of links; it redirects traffic. This flexibility reduces time wastage and improves the results achieved

.

Keys benefits like centralized control, programmability, and flexibility.

Software-Defined Networking (SDN) offers several key benefits that enhance network management, scalability, and agility:

- **Centralized Control**

Simplified Management: SDN brings control of the network into a single point where the administrator can have full control of the network. This makes the configuration, monitoring, and fault finding of the network devices easy and also reduces the complexity of the operation in the network.

Consistent Policy Enforcement: Some of the benefits that accrue from central control include the following: when it comes to applying security rules or traffic management techniques on the network, they are applied consistently across the network hence less chance of developing or implementing faulty policies.

- **Programmability**

Automation: SDN makes the deployment and functioning of the network configurable because decisions are made, scripts written or program applications used in carrying out functions on the network. This in turn minimizes the role of operators in network deployment while also shortening the time required in network deployment and coming up with a lower operational cost.

Customizable Network Behavior: The premeditated programmability of these interfaces makes it possible for SDN to fine-tune a network according to business-demanded designs. Here at the network layer, the service that is offered to clients can be set up, altered, or even eliminated to suit the different loads or applications being used.

- **Flexibility**

Dynamic Adaptation: The main advantage SDN has is the ability to implement changes in response to network conditions at the same time. From merely redirecting traffic because of heavy traffic to changes in policies of the network, SDN gives networks the ability to adapt flexibly.

Support for Virtualization: Software-defined networks allow the creation of virtual networks based on the physical network infrastructure while offering the possibility of traffic management of the different kinds of networks. This is particularly important, especially in cloud structures, where it is possible to need several virtual networks at once.

These benefits include centralized control, programmability, and flexibility all make SDN instrumental in modern network management that will enable organizations to develop efficient, scalable, and responsive networks.

CURRENT ADOPTION AND USE CASES

Software-defined networking or SDN is quickly becoming popular among various businesses especially in cloud computing, data centers, and telecommunications because of its potential to improve network performance, capacity, and elasticity.

Here are some key adoption trends in these sectors:

- **Cloud Computing**

Integration with Cloud Services: Today, SDN is an important way to penetrate cloud networks as it is used for cloud services management and for the effective functioning of complex network structures. This is how cloud providers leverage SDN, making it possible to automate network allocation and aim at security while ensuring that there are virtualized and elastic services.

Multi-Cloud and Hybrid Environments: As companies are moving to Muli-Cloud and Hybrid Cloud, SDN becomes immensely important for networking connections between different clouds. It supports to establishment of coherent policies across the network and adjusts workloads between the own data centers and the cloud.

- **Data Centers**

Automation and Orchestration: In data centers, SDN is being used to enhance the automation procedures on the network and limit manual interactions thereby decreasing network downtime. It helps orchestration platforms to gain better control over the network resources, to address the dynamic requirements of contemporary applications and services.

Scalability: Large-scale data centers benefit from SDN’s capacity to scale network resources on demand. It enables the data centers to quickly grow the total available infrastructure to accommodate the traffic loads and distribute the loads uniformly across the servers and the storage devices.

- **Telecommunications**

5G and Network Slicing: SDN can be seen as a crucial element of 5G networks because it is specifically used for creating a network slicing technology, which consists of the possibility of building multiple different networks based on one single physical network. This capability is instrumental for services with dissimilar service levels, for instance, low and stable latency for IoT devices or high bandwidth for video streaming.

Cost Efficiency and Agility: SDN assists the telecom operators in minimizing the operational expenses due to which the activities in the networks are automated and service organizations can deliver services efficiently. It is also important to quickly react to the emerging market requirements, for instance, with the need for new services like IoT or eMBB.

Here are the examples of use cases for Software-Defined Networking (SDN):

- Network Automation
- Data Center Automation
- Service Provisioning
- Traffic Management
- Dynamic Traffic Routing
- Alongside quality of Service (QoS) Management
- Network Security
- Micro-Segmentation
- DDoS Mitigation
- Network Slicing
- 5G Networks
- Private Networks
- Edge Computing
- Content Delivery Networks (CDNs)
- IOT and Smart Cities.

TECHNICAL CHALLENGES

One of the primary technical challenges in network scaling is bandwidth management. As your business grows and more devices and applications connect to the network, the demand for bandwidth increases significantly. Without proper management, this can lead to slow connections and bottlenecks.

Latency is another critical technical issue. Latency refers to the delay in data transmission, which can be exacerbated as networks expand. High latency can affect everything from your voice calls to real-time data processing.

These technical challenges impact efficiencies, frustrating users, hindering productivity, and losing business revenue. Integration with legacy systems presents yet another technical challenge. If your growing business still relies on older systems that were not designed with modern scalability in mind, integrating new scalable solutions can be complex. These processes can be time-consuming, often requiring significant adjustments and custom configurations.

Interoperability with existing network infrastructure.

Interoperability improves the customer experience. When *network infrastructure* is interoperable, it enables different systems and devices to work together seamlessly. This means that customers can easily access the information and resources they need, without having to worry about compatibility issues. For example, an interoperable *network infrastructure* can enable customers to access their account information and make payments through different channels, such as *mobile devices* and computers.

Interoperability is essential for network infrastructure. It enables different systems and devices to communicate with each other seamlessly, leading to increased efficiency, cost savings, improved scalability, increased innovation, and improved customer experience. When choosing a network infrastructure solution, it is important to consider interoperability as a key factor in the decision-making process.

EMERGING AREAS OF INTEREST

1. A typical IIoT network architecture.

In IIoT networks, the frequent and synchronous industrial service requests are compute intensive, heterogeneous, and high-dimensional. The new intelligent manufacturing technologies involve high-precision industrial manufacturing, processing and maintenance operations requests, which result in delay-sensitive operations – especially with respect to real-time surveillance, computation and cooperation among the different intelligent agents. In addition, different industrial applications are associated with different response time thresholds, representing the different delay-sensitivity requirements. For instance, compared with industrial data backup services, the data flow associated with industrial fault detection presents high degree delay-sensitivity, it requires the network administrator to acquire the entire network states in real-time to determine comprehensive TE strategies according to the features of each industrial application, such that the Quality of Experience (QoE) of each industrial service can be satisfied. Meanwhile, the openness and heterogeneity of IIoT has led to the exposure of massive security holes, making the IIoT networks are vulnerable to various categories of network attack, e.g., DDoS, Probing, R21, U2R, virus, worm, etc. Normally, the attackers falsify the features of abnormal traffic similar to normal traffic, which makes it impossible for network devices to detect the anomaly flow in IIoT network. Further, with the rapid development of IIoT, the category of network flow is becoming more and more diversified. And, different categories of IIoT services produce different flow patterns, making the entire network flow varies over time. This results in the outdate of the anomaly detection models, thus misreporting new flow patterns or failing to detect abnormal traffic in IIoT networks. Hence, it is a non-trial task to improve both the Security and Functionality of IIoT Network. The Software-Defined Networking (SDN) is a recent paradigm that decouples the network control plane from the data plane (short for data forwarding plane). Different from the traditional computer networks, the switches in the data plane are only in charge of delivering the industrial data according to the policies determined at the control plane, while the SDN controller in the control layer functions as the network operator, and aims at managing each switch in the data layer through dedicated control standards.

2. Background and current research position

Recently, diverse research on IIoT networks has been undertaken, especially within the topics of protocols, security, communication technology, architectures, etc. IIoT networks are markedly different from the traditional computer networks. The scale of IIoT networks is usually very limited because they are often deployed to guarantee data delivery in specific domains. This is different from the computer networks, which transfer various categories of multi-media data flow, each of which has different demand on QoS metrics – e.g., security requirement, delay, jitter, data loss, etc. Further, in IIoT networks most of the data is highly sensitive to delays, and one main target is to guarantee that the delay constraints of the data transmission process are satisfied. Thus, different from the traditional computer networks based on CSMA/CD, the MAC layer of the wired IIoT networks is usually built by way of the Rapid Ring protocol. Since the data in the IIoT network typically corresponds to industrial manufacturing services, the data delivery requires to be guaranteed under high-level security policies. In many industrial fields, e.g., coal mining industry, wind power generation, petrochemical industry, etc., the industrial sensors are distributed in unserviced areas, and the sensing information is collected and delivered by open wireless tunnels. This make the industrial sensors not only easy to be stolen and captured, but also vulnerable to eavesdropping. And the traditional security approaches, e.g., trust mechanism, identity authentication, information encryption, etc., cannot satisfy the requirement of industrial applications. Thus, efficient and secure data delivery has to be guaranteed in IIoT networks.

With the widespread implementation of IIoT applications, IIoT networks are evolving gradually towards wired/wireless hybrid network architectures. Presents several candidate protocols for IIoT network communications – ZigBee, WirelessHART, ISA100.11a, WiFi, Bluetooth, LoRA are wireless communication protocols, while PROFINET, Ethernet POWERLINK, DeviceNET, Modbus, and CAN are wired protocols. The recent successful commercialization of 5G brings forth another innovation opportunity, upgrading the IIoT networks to High Bandwidth and Low Latency (HBLL). IIoT integrating 5G technology is more capable of real-time industrial data sensing, supporting timely data processing or analysis at the data computing center.

3. AI-SDIN architecture

As a revolutionary technique to improve the scalability and robustness of network functionality, the SDN paradigm has been widely used in multiple types of networks – e.g., traditional computer networks, multi-agent networks, underwater acoustic sensor networks, etc. SDN utilizes standardized protocols OpenFlow, ForCES, POF etc. between network control and packet forwarding components. By SDN, the network control unit can be intensely decoupled from the network data delivery devices, such as the router, leading to a centralized network control plane, i.e., the SDN controller. Breaking away with the traditional manner of individually configuring network devices, SDN controllers can support exact network monitoring and adaptively determine/deploy network policies, according to users' requirements and the network states.

To yield intelligent policy decisions, AI-enabled computing elements can be deployed on the SDN controller, so that smart network control and functions are executed. For instance, AI algorithms can assist in identifying and isolating malicious attacks in a software-defined network. Thus, SDN applications gives to IIoT networks an scalable platform that can deploy complex network policies, while AI, functioning as *machine intelligence*, can compute smart and self-adapting network policies.

SIGNIFICANCE OF THE STUDY

1. CONTRIBUTION TO THE FIELD

Filling Knowledge Gaps: This research is novel in determining problem areas in SDN that are hindering its deployment and expands in the areas of scalability, security, and interoperability. In addressing these issues, the study contributes to the existing body of knowledge and lays down the groundwork for subsequent research endeavors.

Practical Applications: The implications of this study are discussed in the following context regarding network administrators, developers, and organizations embracing or seeking to embrace SDN. This study helps enrich the literature about the different practices relating to network management and assists in providing newer and better solutions for the betterment of network performance.

2. **INFLUENCE ON FUTURE DEVELOPMENTS:**

Guiding Future Research: The research identifies the gaps that should be addressed and further examined, for example, SDN's ability to address cyber threats or integration of the SDN with newer technologies like 5G and Edge computing. These interpretations will guide the course of subsequent studies thus enabling the continued evolution of SDN about progression in technology and other requirements of the sector.

Driving Innovation: As a result, this research identifies the key issues of SDN and contributes to the systems' evolution by encouraging the creation of unique solutions and devices. These developments could further extend various aspects of networks such as stability, scalability as well as flexibility as a key drivers to the progression of SDN across different industries.

Therefore, this research is important in contributing to the development of SDN specifically by solving existing problems while also establishing the foundation for possible enhancements. The results may have profound implications on the path of SDN evolution to make it a better solution to contemporary networking challenges.

SIMULATION, DEVELOPMENT, AND DEBUGGING TOOLS

The development of SDN has seen the advent of several key simulation and emulation test beds to carry out feasibility studies and introduce new protocols and services. The set of tools available for the purpose can be broadly divided into four categories:

- Simulation and emulation platforms
- Software switch implementations
- White box solutions
- Debugging and troubleshooting tools.

A summary description of major utilities within each category and their description are given in Table 5. An overview of tools is given below. Simulation and Development Platforms. Among the emulation tools, Mininet is the most prominent. The platform allows an entire network based on OpenFlow to be emulated over a single cluster of machines. The distribution of Mininet nodes and links over a cluster of machines utilizes the resource of each machine, adding scalability to emulate larger networks requiring more computation and communication bandwidth than available on a single Mininet server. Mininet simplifies the development and deployment of new services by providing a software platform to create virtual machines, hosts, and network switches connected to an in-built (ovs-reference) or user-defined controller for testing purposes. At the access level, white box switches may be utilized to offer functionalities from wireless network control to LAN traffic forwarding, orchestrated by the SDN controller. Customizable feature adoption allows administrators to utilize the white box devices in multiple settings as dictated by the SDN applications. White box solutions, however, require a significant deal of expertise from administrators to accurately configure the devices for subsequent use, sometimes without sophisticated manufacturer after-sales support. Several early deployments of white box switches are therefore seen in relatively large service providers, leveraging existing network management experience to offer greater innovation using SDN programmability with commodity hardware. Open source initiatives such as the Open Network Install Environment (ONIE) further enable administrators to install any network operating system on hardware devices with a great deal of automation, essentially enabling management of white box switches which is quite similar to servers. Branded white box switching gear without a default operating system is available from several manufacturers. The software to be loaded (operating system) is available from either relatively new start-up companies to more established network solution providers.

RESULTS

In the course of our study on Software-Defined Networking (SDN), we found out several tools and platform that are important in mitigating challenges experienced in implementation of SDN and also in finding research gaps. The highlights of the tools and platforms identified in the present study are presented below based on their efficiency in solution of main SDN problems and opportunities for further research.

1. Simulation and Emulation Platforms

Mininet: Mininet was found to be a critical resource in the modeling and prototyping of the SDN architectures. Its capability to mimic whole networks using Open Flow on a single, or a set of, machines has proved very useful when testing scalability for example. The fact that nodes and links can be distributed between many machines also enables one to simulate more nodes and links than are available in a small lab; this addresses the issues of computing and bandwidth constraints. The use of Mininet makes it very easy in the development and testing of new SDN services as well as deployment without any risk involved since it is not a real environment.

2. Software Switch Implementations

Open vSwitch (OVS): Open vSwitch has been revealed as one of the most used software switch implementation in the SDN network. Meanwhile, its compatibility with Mininet and other emulating platforms helps to create a rather free environment for SDN controllers to engage with virtual network components. Due to its capability to work across different conditions of the network, OVS is a key in designing and implementing experimental SDN protocols and services. Its deployment also underlines its role in the SDN environment especially in the environments that require quick deployment and the ease of integration of various network functionalities.

3. Overall Effectiveness and Opportunities

The tools discussed in this paper as a whole help to mitigate the key problems that may occur in SDN or the implementation of that concept, including concerns related to its scaling, flexibility, and individual adaptability. But they also marked areas that require the additional research, particularly the white box solutions more friendly for the users and the new efficient means of the program debugging. The paper highlights that one cannot stop investing in developments of new SDN tools and platforms to enhance the possibilities of SDN and handle the difficulties that impede its implementation.

4. Research Opportunities

The paper outlines several areas for future research which include: A more complex simulation and emulation of real-life SDNs; improving white box solutions for application by other than experts; and the development of finer tools for identifying problems with SDN architectures.

The outcomes acquired in this paper shed the light on the contemporary state of SDN tools and platforms' characteristics and influence, as well as their further perspectives. The results presented in the paper provide a basis for further studies focused on the elimination of the barriers to the effective implementations of the SDN in today's network environment.

DISCUSSION

In SDN the network conditions are much more dynamic because of facilitated, frequent software installation and changes. As a consequence, this is also more likely to create situational issues, which are not predicted in the design of management systems. To deal with unexpected and temporary conditions, such as debugging a newly installed software protocol, tools need to be available for fast creation of on-demand management applications taking advantage of information available from SDN planes and interfaces. Adequate situation management also encompasses fulfilling the requirement of monitoring and visualization of SDN. In general, the SDN controller fills the flow table with the maximum number of possible flows. In reactive mode, flow setup is performed when a packet arriving at the switch does not match any of the switch entries. Then the controller will decide how to process/handle that packet, and the instructions will be cached onto the switch. As a result, reactive flow-setup time is the sum of the processing time in the controller and the time for updating the switch as the flow changes. Therefore, flow initiation adds overhead that limits network scalability and introduces reactive flow-setup delay. In other words, a new flow setup requires a controller to agree on the flow of traffic, which means that every flow now needs to go through the controller, which in turn instantiates the flow on the switch. However, a controller is an application running on a server OS over a 10 GB/sec link (with a latency of tens of milliseconds). It is in charge of controlling a switch which could be switching 1.2 TB/sec of traffic at an average latency of 1μs. Moreover, the switch may deal with 100K flows, with an average of 30K being dropped. Therefore, a controller may take tens of milliseconds to set up a flow, while the life of a flow transferring 10MB of data (a typical Web page) .

CONCLUSION

Despite being a relatively recent networking paradigm, the importance of SDN is already evidenced by the emergence of many start-up companies and foundations, the interest of researchers, and the support of major Internet players. Much has been recently discussed on taking SDN as a tool to simplify some classical network management issues. However, in this article we took a different perspective by analyzing the management necessities that did not exist before the inception of SDN. Initially we revisited the discussions about the definition of SDN regarding concepts such as separation of planes and the actual implementation of a control plane. We approached SDN from a slightly different angle than many other authors, in which we emphasize the fact that SDN is essentially about abstracting network logic from hardware implementation to software. We also provided evidence that SDN currently overlaps with other emerging related concepts, such as Network Functions Virtualization and Software Friendly Networks. Furthermore, we included in the SDN architecture a conceptual plane dedicated to implementing management functions, either in a centralized or distributed manner

Finally, we established a set of challenges that we consider a fundamental contribution to encourage future investigations regarding SDN management. We envision mainly the resurgence of traditional network management concepts, such as autonomic/self-management and policy based network management. Moreover, we believe in the empowerment of situational management, for example, based on mash up-oriented technologies. Most importantly, we understand that SDN represents a landmark: for the first time in decades we are witnessing computer network development happening outside private industry boundaries. In these times of "networking democracy" a crucial opportunity presents itself to address management requirements, and to avoid the recurrent mistake of patching management solutions after other concepts are already mature.

REFERENCE

Santhosh Katragadda

Big Switch Networks, FloodLight OpenFlow Controller, http://www.projectfloodlight.org/floodlight/,2013

Bae, H., "SDN Promises Revolutionary Benefits, but Watch Out for the Traffic Visibility Challenge," http://www.networkworld.com/news/tech/2013/010413-sdn-trafficvisibility-265515.html

Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D. C., & Gayraud, T. (2014). Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, *75*, 453-471.https://doi.org/10.1016/j.comnet.2014.10.015

Jammal, M., Singh, T., Shami, A., Asal, R., & Li, Y. (2014). Software defined networking: State of the art and research challenges. *Computer Networks*, *72*, 74-98. https://doi.org/10.1016/j.comnet.2014.07.004

Al-Shareeda, M. A., Alsadhan, A. A., Qasim, H. H., & Manickam, S. (2024). Software defined networking for internet of things: review, techniques, challenges, and future directions. *Bulletin of Electrical Engineering and Informatics*, *13*(1), 638-647. https://doi.org/10.11591/eei.v13i1.6386

Imran, Ghaffar, Z., Alshahrani, A., Fayaz, M., Alghamdi, A. M., & Gwak, J. (2021). A topical review on machine learning, software defined networking, internet of things applications: Research limitations and challenges. *Electronics*, *10*(8), 880. https://doi.org/10.3390/electronics10080880

Bakhshi, T. (2017). State of the art and recent research advances in software defined networking. *Wireless Communications and Mobile Computing*, *2017*(1), 7191647. https://doi.org/10.1155/2017/7191647

Li, Y., Su, X., Ding, A. Y., Lindgren, A., Liu, X., Prehofer, C., ... & Hui, P. (2020). Enhancing the internet of things with knowledge-driven software-defined networking technology: Future perspectives. *Sensors*, *20*(12), 3459. https://doi.org/10.3390/s20123459

Hussain, M., Shah, N., Amin, R., Alshamrani, S. S., Alotaibi, A., & Raza, S. M. (2022). Software-defined networking: Categories, analysis, and future directions. *Sensors*, *22*(15), 5551. https://doi.org/10.3390/s22155551

Guesmi, T., Kalghoum, A., Alshammari, B. M., Alsaif, H., & Alzamil, A. (2021). Leveraging software-defined networking approach for future information-centric networking enhancement. *Symmetry*, *13*(3), 441.https://doi.org/10.3390/sym13030441

KEYWORDS

Software-Defined Networking (SDN)
SDN Challenges
SDN Research Opportunities
Traditional Networking
SDN Scalability & Security
Mininet SDN
Future of SDN
SDN and 5G

SDN Deployment Strategies
Network Management with SDN
Research Opportunities for Future Interest