Software-Defined Networking: Self-Healing Topology Discovery Protocol for Software Defined Networks

T. Vamshi Mohana [1], Dr. Baddam Indira [2]
[1]Research Scholar, Career Point University
[2]Research Supervisor, Career Point University

***Abstract* - 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. However, both concepts can be merged and have the potential to mitigate the challenges of legacy networks. In this paper, our aim is to describe the benefits of using SDN in a multitude of environments such as in data centers, data center networks, and Network as Service offerings. We also present the various challenges facing SDN, from scalability to reliability and security concerns, and discuss existing solutions to these challenges.**

***Index Terms* - Software- Defined Networking, OpenFlow, Datacentres, Network as a Service, Network Function Virtualization.**

1.INTRODUCTION

Today's Internet applications require the underlying networks to be fast, carry large amounts of traffic, and to deploy a number of distinct, dynamic applications and services. Adoption of the concepts of "inter-connected data centers" and "server virtualization" has increased network demand tremendously. In addition to various proprietary network hardware, distributed protocols, and software components, legacy networks are inundated with switching devices that decide on the route taken by each packet individually; moreover, the data paths and the decision-making processes for switching or routing are collocated on the same device. This situation is elucidated in Fig. 1. The decision-making capability or network intelligence is distributed across the various network hardware components. 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 [1, 2]. 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 [2]. 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 [2, 3].

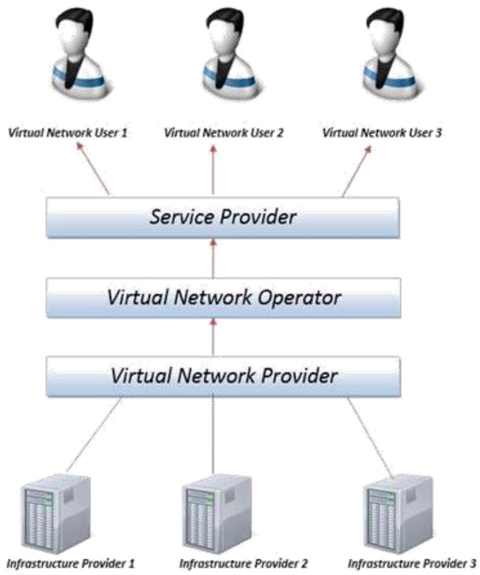

Figure 1: Inflexible Legacy Infrastructure

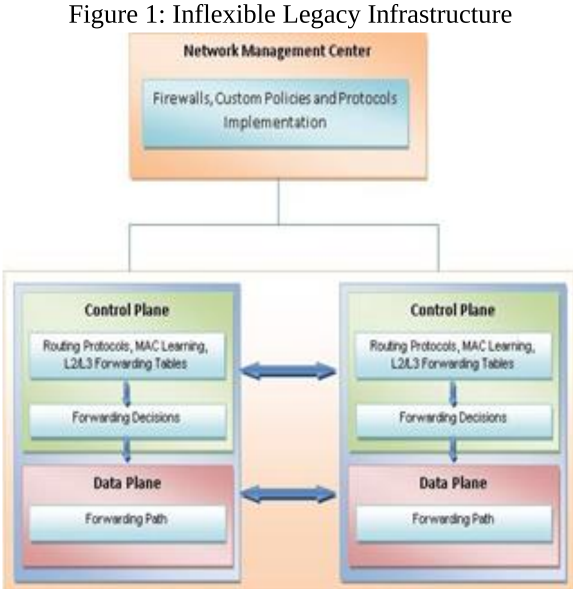

Figure-2: General Network Virtualization Architecture

1.1Benefits of Network Virtualization

Some of the key benefits offered by network virtualization are mentioned below [18, 19]:

1. Co-existence of Dissimilar Networks

Network virtualization makes it possible to create multiple virtual networks on the same physical hardware. However, these virtual networks can be isolated from other existing virtual networks. This isolation can be used as a tool in the deployment of networks using different or even incompatible routing protocols.

2. Encouraging Network Innovation

Like SDN, network virtualization can be used to encourage innovation in the networking domain. The isolation that can exist between two virtual networks can be used to create separate domains for production traffic and test traffic. This isolation guarantees that a malfunction experiment will not affect production traffic.

3. Provisioning of Independent and Diverse Networks

NV deploys packet handling, quality of service (QoS) and security policies to configure network operations and behaviors. This configuration allows the categorization of different networks based on their services, users and applications.

4. Deployment of agile network capabilities

The inclusion of agile facilities into the current network improves the data transport efficiency and provides robust network. With the agile manner, NV allows the integration between legacy and advanced networks. Also, it enables migration from legacy systems into advanced ones in an agile manner.

5. Resource Optimization

The dynamic mapping of multiple virtual network nodes to the physical substrate ensures that the network hardware is utilized up to capacity. This approach cuts down on hardware costs and delivers additional profit to the infrastructure provider.

6. Deployment of Distinct Network Services

Network services such as wireless local area networks (WLANs) and Intranet require specific network architectures. In addition, a multi-national corporation might need to offer distinct services to its employees. This can add complexity to the existing overlay network. Network virtualization can help alleviate these problems by deploying such services in separate virtual networks.

2.NETWORK FUNCTION VIRTUALIZATION

As for the ambiguity between the concepts of network function virtualization (NFV) and SDN, it is necessary to take advantage of the definitions and benefits of both technologies.

2.1 Definition of NFV

Expansion of the deployment of various applications and network services induced service providers to come up with the concept of NFV. Therefore, they established a European telecommunication standard institute (ETSI) Industry Specification Group for NFV. The group defined the real concept of NFV together with its requirements and architecture.

NFV decouples network functions, e.g., firewalls, domain name service (DNS), and caching, from dedicated hardware appliances and entrusts them to a software-based application running on a standardized IT infrastructure, high-volume servers, switches, and storage devices. The interesting feature of NFV is its availability for both wired and wireless network platforms. NFV reduces capital expenditures (CAPEX) and operating expenditures (OPEX) by minimizing the purchase of dedicated hardware appliances, as well as their power and cooling requirements. Virtualization of network functions enables fast scale-up or scale-down of various network services and provides agile delivery of these services using a software application running on commercial off-the-shelf (COTS) servers.

convergence architectures to fix the fractures in the DCN infrastructure, these solutions do not address the problems in heterogeneous networks. Nevertheless, the software-defined network paradigm is a promising solution to solve these challenges in DCN setups.

3.SDN DEPLOYMENT IN DCNS

In SDN Open Flow based networks, the virtual network segments are centrally configured, and network security is simplified by directing flows to security policy services. Moreover, the central controller transforms the core and aggregation devices into a "high-speed transport backplane" [20]. The controller can provision a new device that is added to the network and allow it to receive the configuration policy when it appears online. Finally, SDN improves DCN infrastructure, its power consumption, and its various metrics. Due to these improvements and modifications, different SDN applications in DCNs have been proposed.

a. Changes in DCN Infrastructure

Automation and virtualization of data-center LANs and WANs has resulted in a flexible and dynamic infrastructure that can accommodate operating-cost challenges. As a result, Vello Systems [22] has proposed an open and scalable virtualization solution that connects the storage and computation resources of the data center to private and public cloud platforms. To facilitate the migration of VMs from their Layer 2 network, Layer 2 was extended across multiple DCs using Layer 3 routing. However, Layer 3 routing introduces challenges in intra-data center connectivity and cannot meet the requirements for VM migration across DCs. Therefore, the proposed solution is based on a cloud-switching system that enables cloud providers and enterprises to overcome the traditional Layer 2 domains, the direct server-to-server connection, and virtual server migration.

Because the switching system supports integration of end-to-end network attributes, its operating system can provide a

Green DCN		
Proposed Solution	Objective	Functionality
OpenFlow platform for energy-aware data center [24]	Provide guidelines for studying energy consumption in DCN elements	• Estimate the minimum power for a given network topology. • Satisfy the traffic conditions and QoS requirements. • Provide a power module in the controller that determines the power state of network elements. • No evaluation of the proposed approach on different network topologies
OpenFlow switch controller (OSC) [26]	Decrease the influence of carbon emissions in the DCs	• Reduce configuration time of network elements. • Enable flexible power management operations based on the programmable controller

The proposed approach predominantly varies from existing work in that it expects to reproduce and keep an organization's geography, instead of simply its network or measurement. This is doable in application settings where hubs can be reproduced and reconnected (for example spare workers in Data Centers; torpid gadgets in sensor organizations; repetitive robots or handhelds in specially appointed versatile organizations). For example, [9] proposed a plan technique for becoming both vigorous and effective onion-like topological constructions. [2] pointed to construct a solid geography dependent on a fractal cell-structure and contrasted it and sans scale organizations, as characterized in [3]. This is corresponding to our methodology in that we mean to recuperate the network geography in the event of disappointment. Different methodologies expected to manage hub disappointments by reconnecting remaining hubs; thus, staying away from network parting into secluded segments, which would render the framework useless. E.g., the recommendations in [5] and [18] trigger hub reconnection to keep up availability when a hub loses a specific number of neighbors; or, to keep up internode distances like the first organization or possibly under a most extreme distance. As in the past, this methodology doesn't recuperate fizzled hubs and consequently doesn't save network geography. [13] proposed a self-recuperating convention for Software Defined Networks (SDNs). It intended to keep a given geography in two stages: 1) utilizing multi-cast for network disclosure and state information assortment; what's more, 2) utilizing an autonomic disappointment recuperation component. Investigations were performed for sans scale organizations. Our methodology goes more profound into investigating and assessing elective information assortment instruments (in light of Trickle and Mobile Agents), to decrease asset overheads. We adjust Trickle and Mobile Agents calculations to give hubs topological information, as important to keep a directed organization geography. Besides, we break down the productivity of these calculations for various geographies, showing that underlying qualities, however regularly overlooked, do matter. Stream is a scaleable and powerful calculation for engendering and keeping up data in low-power, lossy organizations (for example remote sensor organizations). It was characterized under the RFC6206 standard [12], with regular applications including traffic timing control, multicast proliferation and course revelation. We embrace Trickle as a gauge for assessing our Mobile Agents approach

4.CONCLUSION

SDN aims to simplify network architecture by centralizing the control-plane intelligence of L2 switching and L3 routing equipment. It also markets network hardware as a product service and forms the basis of network virtualization. The generalized SDN architecture consists of the SDN controller and SDN-compatible switches. Because SDN makes it possible to build programmable and agile networks, academic researchers and network engineers are exploiting its flexibility and programmability to generate strategies that simplify the management of data-center LANs and WANs and make them more secure. Besides, SDN supports NaaS, the new Internet based model that acts as a link between cloud computing and SDN. While SDN manages forwarding decisions and network administration, NaaS will provide packet-processing applications for cloud tenants. In addition, researchers are proposing various SDN prototypes that will serve DCNs, wireless networks, enterprises, and campus networks. Despite all the promising opportunities that accompany SDN, it encounters certain technical challenges that might hinder its functionality in cloud computing and enterprises. Therefore, IT organizations and network enterprises should be aware of these challenges and explore the functionality of the SDN architecture to counter these criticisms.

REFERENCES:

[1] Amazon. [n.d.]. Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region. https://aws.amazon.com/es /message/41926/.

[2] Ya Nan Bai, Ning Huang, Lina Sun, and Lei Wang. 2019. Reliability-based topology design for large-scale networks. ISA Transactions 94 (2019), 144–150. https: //doi.org/10.1016/j.isatra. 2019.04.004.

[3] Albert-László Barabási and Eric Bonabeau. 2003. Scale-Free Networks. Scientific American 288, 5 (May 2003), 60–69. https://doi.org/10.1038/ scientificamerican0503-60.

[4] Zhenhao Chen, Jiajing Wu, Zhihai Rong, and Chi K. Tse. 2018. Optimal topologies for maximizing network transmission capacity. Physica A: Statistical Mechanics and its Applications 495

(2018), 191–201. https://doi.org/10.1016/j.physa. 2017.12. 084.

[5] Lazaros K. Gallos and Nina H. Fefferman. 2015. Simple and efficient self-healing strategy for damaged complex networks. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 92, 5 (2015). https://doi.org/10.1103/PhysRevE. 92.052806 arXiv:1511.06729.

[6] Mohsen Ghaffari and Bernhard Haeupler. 2013. Near optimal leader election in multi-hop radio networks. In Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms. SIAM, 748–766.

[7] Volker Grimm, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske, and Steven F. Railsback. 2010. The ODD protocol: A review and first update. Ecological Modelling 221, 23 (2010), 2760–2768. https://doi.org/10.1016/j. ecolmodel.2010.08. 019.

[8] Volker Grimm, J Gary Polhill, and Julia Touza. 2013. Documenting social simulation models: the ODD protocol. Simulating social complexity: a handbook (2013), 117–34.

[9] Yukio Hayashi. 2016. Spatially self-organized resilient networks by a distributed cooperative mechanism. Physica A: Statistical Mechanics and its Applications 457 (2016), 255–269. https://doi.org/10.1016/j.physa.2016.03.090 arXiv:1603.08329.

[10] Fabian Kuhn. 2020. Faster Deterministic Distributed Coloring Through Recursive List Coloring., 1244–1259 pages. https://doi.org /10.1137/1.9781611975994.76.

[11] Philipe Lalanda, Julie A Mccann, and Ada Diaconescu. 2013. Autonomic Computing: Principles, Design and Implementation. Springer.

[12] Philip Levis, Thomas Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. 2011. The trickle algorithm. Internet Engineering Task Force, RFC6206 (2011).

[13] Leonardo Ochoa-Aday, Cristina Cervello-Pastor, and Adriana Fernandez 2018. Self-Healing Topology Discovery Protocol for Software Defined Networks. IEEE Communications Letters 22, 5 (May 2018), 1070–1073. https://doi.org/10.1109/LCOMM.2018.2816921.

[14] Luiz A. Rodrigues, Elias P. Duarte, and Luciana Arantes. 2018. A distributed k-mutual exclusion algorithm based on autonomic spanning trees. J. Parallel and Distrib. Comput. 115 (2018), 41–55. https://doi.org/10.1016/j.jpdc.2018.01.008

[15] Arles Rodríguez, Nathaly Botina, Jonatan Gómez, and Ada Diaconescu. 2019. Improving data collection in complex networks with failure-prone agents via local marking. Journal of Intelligent and Fuzzy Systems 36, 5 (May 2019), 5081–5089. https://doi.org/10.3233/JIFS-179053 .

[16] Arles Rodríguez, Jonatan Gómez, and Ada Diaconescu. 2020. A decentralised self healing approach for network topology maintenance. Autonomous Agents and Multi-Agent Systems 35, 1 (2020), 6. https://doi.org/10.1007/s10458-020-09486-3.

[17] Remco Van Der Hofstad. 2016. Random Graphs and Complex Networks Vol. I. Available on http://www.win.tue.nl/rhofstad/NotesRGCN.pdf I(2016).http://www.win.tue.nl/{~}jkomjath/Note s RGCN2013may.pdf

[18] Tianyu WANG, Jun ZHANG, and Sebastian WANDELT. 2017. Exploiting global information in complex network repair processes. Chinese Journal of Aeronautics 30, 3 (2017), 1086–1100. https://doi.org/10.1016/j.cja.2017.03.007