Software-Defined Networking: Open-Source Alternatives for Small to Medium Sized Enterprises

THORNLEY, Paul and BAGHERI, Maryam

Software-Defined Networking: Open-source alternatives for Small to Medium Sized Enterprises

Paul Thornley[1], Maryam Bagheri[2]
[1] Sheffield Hallam University b9017293@my.shu.ac.uk
[2] Sheffield Hallam University maryam.bagheri@shu.ac.uk

Abstract. SDN Networking is a new and emerging technology and is receiving significant considerations within organizations. Previous research predominantly emphases on SDN deployment with large Local Area Networks (LAN's) and its use in corporate Wider Area Networks. SDN adoption within small networks is an area of research that has not been explored in any detail. This paper aims to analyze a market-leading SDN solution – Cisco DNA Centre- and identify key technologies for utilization with SME-based networks. Furthermore, this research aims to provide open-source SDN solutions suitable for Small to Medium-sized networks and show how those identified technologies could be developed and deployed within Open-source solutions. This research shows that Open-source SDN controllers are a viable and deployable solution within small to medium-sized business networks.

1 Introduction

The demands made on today's business computer networks is arguably at its highest, more than it has ever been. Key stakeholders and decision-makers within organizations continually face challenges when considering the technologies and concepts available in delivering efficiency and effectiveness within IT infrastructure. Organizational budgets and technological complexities are often in competition within key stakeholder considerations, where any introduction of technologies with increased complexity, often have a significant impact on available budgets [23]. Additionally the move towards concepts such as the Internet of Things (IoT), borderless networks, and cloud computing has increased the need for the underlying infrastructure to adapt and evolve in response to business demand. Whilst there are many techniques and guidelines outlining design methodologies to provide scalability and expansion of networks [3], these often do not consider how networks respond to demand dynamically. This inability to evolve its physical infrastructure poses challenges when introducing new technologies into infrastructure and is referred to as 'Internet ossification' [18]. As a response, organizations have moved towards network programmability and automation to address the issue of 'Internet Ossification' and provide network infrastructure that responds dynamically to network needs and demands. One such concept is the paradigm of Software Defined Networking and its key objective of introducing network automation and programmability within IT infrastructure.

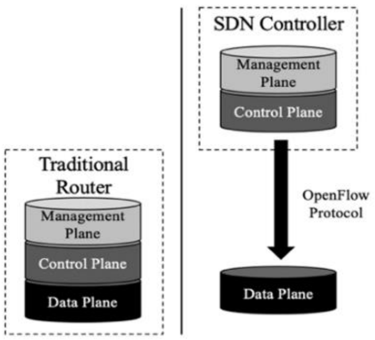
According to current figures, the Software Defined Networking market was worth $8 Billion (USD) in 2019 and is expected to grow 40% and be worth an estimated $100 Billion (USD) by 2025 [2]. The operational benefits of introducing SDN based technologies within a networked infrastructure has been widely suggested and supported by the projected increase in the market value of SDN technologies. Several studies suggest that the implementation of SDN technologies provides significant improvements in network scalability, elasticity, network management, and response to demand in comparison to using traditional networking hardware and concepts [9], [12]. This makes SDN technologies a significant consideration for organizational key stakeholders when considering the re-development or implementation of networked infrastructure. However, market-leading SDN solutions such as Cisco DNA Centre are significantly costly and often complex to implement. With smaller profit margins and lower IT budgets, this could result in SME businesses being unable to introduce SDN technologies within their infrastructure due to SDN solution costs.

Currently, there is diminutive research and documentation available that provides affordable suggestions and guidance for the development and deployment of SDN based technologies that are purposely targeted at SME businesses.

To address this, this paper aims to ascertain how SDN technologies are applied to smaller networks typically found within SME IT infrastructure. We will analyze and identify key technologies within a market-leading Software Defined Networking solution and compared them against open-source or lower cost SDN alternatives. Prototype networks to demonstrate the viability of SDN within smaller SME networks will be created in EVE-NG simulator.

2 Literature Review

SDN solutions today are becoming a major consideration for organizations and key stakeholders within the business. Limitations within traditional networking infrastructure are one of the driving factors that has contributed to the shift towards Software-Defined Networking concepts. Lack of options within the programmability and automation of networking nodes often results in un-responsive networks, poor load balancing, and require configuration to be conducted through different vendor-specific configuration interfaces [13]. In the mid-2000s researchers at Stanford University sought to address this issue by developing a method for logically centralizing the control and data planes of networking devices [5].



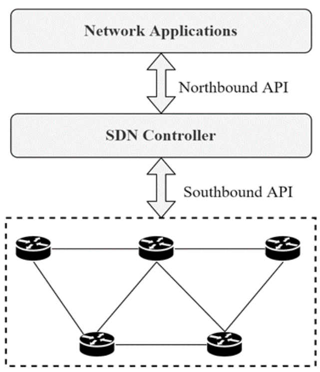


Fig. 1 Traditional v SDN - Control and Data Planes

Fig. 2 SDN Architectural

Fig. 1 demonstrates the abstraction of the control plane and links to a centralized SDN controller. The data plane of networking nodes remains within the device itself, resulting in the device being only concerned with the operation of forwarding data. This centralization of the control plane within the SDN controller provides opportunities to create a holistic perspective of network performance and control plane-based configurations [13], [20]. To provide network software and hardware abstraction, the centralized SDN controller uses Northbound (NBI) and Southbound (SBI) Application Programmable Interfaces [13]. Fig. 2 shows the placement of the controller with the NBI providing the link to software and the SBI providing communication to network devices. By creating this abstraction of software and hardware within the paradigm of networking, could arguably increase the ease of management and configuration of networking devices, even under network growth and expansion of the infrastructure. Since SDN initial conception, several research projects were conducted aiming to analyze and identify the key benefits of introducing SDN into an organizations network. A study conducted in 2014, introduced SDN networking gradually into a large network and operated SDN alongside legacy networking devices [14]. In their study, it was found that using a centralized controller achieved faster network convergence times after link failures identified through the Spanning Tree Protocol (STP). Also, faster download speeds through improved multi-path forwarding of data as compared to the traditional legacy devices. These benefits are further supported in a study conducted in 2014, where five key SDN deployment benefits were identified. The author in [10] argues the dynamic nature of SDN through its response to networking data and the ability to make changes in real-time in response to returned data. Furthermore, they argue the elasticity of SDN technologies where additional nodes, links and routing decisions can be either be created or removed in response to networking needs.

2.1 SDN Architecture in Smaller Networks

A large amount of research has been focused on the design, deployment, and implementation of SDN technologies within bigger networking architectures such as large corporations and datacenters. Studies such as the development of a large-scale SDN testbed in [8] and Google's B4 project [9] provide evidence in support of the benefits inherited through the deployment of SDN across multiple large campus networks and datacenters. But they fail to suggest how these concepts could be introduced into smaller network architectures where skills, resources, and budgets are significantly limited. Without the research of SDN technologies focused on the deployment within smaller networks could provide difficulties in organizations deciding to move towards SDN architectures.

Research performed in 2014 suggested a methodology for the deployment of low-cost hardware and open-source software to create a small test-bed network. This was achieved by introducing the Floodlight SDN Controller to a centrally located Raspberry Pi device and operating as the main SDN controller. Besides, the switching fabric of the network consisted of further Raspberry Pi devices operating as Open Virtual Switches (OVS). This SDN deployment on Raspberry Pi devices provided systems such as Quality of Service (QoS), Load Balancing, and topology overviews at a very low cost in comparison to traditional networking hardware [10]. The deployment of SDN based technologies within this study evidenced the possibility of applying this concept of networking within smaller SME-based topologies.

The author in [6] developed the 'Neto-App' to address the shortfalls found within SDN technologies in supporting smaller academic and business networks. The 'Neto App' applied the concept of an underlay network consisting of layer 3 routing and on overlay network providing network segmentation by applying a technology called Virtual Extensible Local Area Networks (VXLAN). They suggested by applying the underlay/overlay concept, an SME organization can deploy an SDN based solution on limited networking hardware and introduce benefits inherited using SDN technologies.

There is strong evidence in support of applying SDN technologies within SME networks. This can bring efficiency, management, development, and scalability improvements throughout all areas of the network. However, SMEs seeking to adopt new technologies within existing infrastructure often experiences barriers such as technical support, poor understanding of new systems, and overall costs [19]. With the increasing costs of ICT based OPEX and limited profit margins within SME's, the ability to fund specialist expertise or staff for SDN development may be limited. Arguably, this could effectively 'price-out' SME businesses from deploying SDN technologies. This could prevent SMEs from experiencing benefits inherited through the introduction of SDN concepts and fall behind competitors that have access to new technologies and larger IT budgets. This provides research opportunities in analyzing key aspects of SDN technologies and identifying solutions that could be potentially implemented within an SME network at little or zero cost.

3 Methodology

This research intends to establish methods of deploying key SDN features identified from a market-leading solution and applying open-source alternatives with an SME. From this, we can formulate the main question: how can open-source alternative SDN technologies provide free or low-cost solutions for small to medium-sized businesses? To provide a solution to this question, first, the key technologies of the Cisco DNA Centre will be identified. By determining the key features of Cisco DNAC which provides the core functionality, we will use previous literature and implement the same features in an open-source alternative suitable for SMEs.

3.1 Cisco DNA Centre

Cisco DNA Centre (DNAC) is an SDN solution for the enterprise networks released by Cisco, the world's largest networking company. Cisco DNAC is a highly featured network controller and management dashboard that automate networks, deploy group-based secure access and network segmentation, provide assurance and can manage heterogeneous network devices by integrating Cisco solutions with third-party technologies. Case studies from Cisco do show that the Cisco DNAC solution can be implemented and introduce benefits within smaller organizations. However, this solution is considerably expensive and SME organizations may find this solution exceeds financial constraints and maintain underperforming networks consisting of traditional networking devices. This would suggest SME's seeking to introduce SDN technologies within their infrastructure would need to consider alternatives due to limited budgets and other financial constraints.

3.2 Key Technologies in Cisco DNA Centre

In Cisco SD-Access architecture, to promote network segmentation across both underlay and overlay fabrics, Cisco DNAC uses the Location Identifier Separation Protocols (LISP) within the underlay fabric and VXLAN technologies within the overlay fabric. By combining the two technologies allows for the distribution of segmented layer 2 LANs networks across a layer 3 routed underlay, increasing network performance by removing issues inherited by spanning tree and link redundancy [7]. Arguably, the use of layer 3 routed underlay devices could provide SME organizations with the ability to easily expand their networks across multiple geographical locations. Furthermore, LISP provides a holistic approach to the domain and network segmentation as opposed to individual devices performing this functionality. This reduces configuration tasks performed by network engineers and developers seeking to implement network segmentation potentially reducing costs within the development.

Whilst it is recognized that Cisco DNAC contains many more functions and scope for SDN development, the purpose of this research is to establish the viability of VXLAN and LISP configurations for use within an SME. A virtualized platform was used to create prototype concept networks consisting of the technologies and the identified configuration methods.

3.3 Open Source SDN Controller for SME-based Network

The controller plays a critical role within the functionality of an SDN-based network and serious considerations should be made on the choice of the controller and its placement. Misconfiguration and deployment of an SDN controller can distribute errors and performance issues throughout an organization's infrastructure [24]. To prevent this, key stakeholders and network engineers must fully consider how SDN is structured and how some of the key programming languages and protocols are applied within the SDN architecture. As such, these considerations may be a deciding factor when choosing an SDN controller within an SME-based network. Table 1 shows the most prominent SDN controllers in use today and suggests the wide variety of programming languages and APIs used within its operation. Of note, the REST API can be utilized within the SBI of most identified controllers and can be supported by any programming language that includes a REST API library. This provides software developers with the opportunity to create network-based applications to configure control plane logic using a variety of programming languages, irrespective of the programming language used to create the controller itself.

Table 1. SDN Controller & programming languages [24], [25]

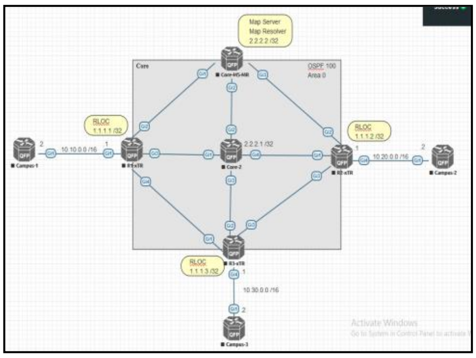
Controller	Programming Language	Northbound APIs
Pox	Python	ad-hoc
Ryu	Python	REST
Nox	C++	ad-hoc
Floodlight	Java	Rest, Java, RPC, and Quantum
Beacon	Java	ad-hoc
OpenDaylight	Java	REST, RESTCONF, XMPP and NETCONF
ONOS	Java	REST and Neutron

Of note is the OpenDaylight SDN controller that provides the largest scope for API development within SDN-based developments. RESTCONF and NETCONF are network-based APIs that are not new to the world of networking. These APIs are often utilized within network programmability applications with many networking device operating systems supporting the use of these APIs [4]. Since these APIs have been widely available outside the paradigm of SDN architecture, many network engineers and developers will already possess knowledge of these APIs. Arguably, since developers would potentially hold the relevant skill sets using APIs such as RESTCONF and NETCONF, this makes the OpenDaylight controller a significant consideration within SDN architecture within SME's.

3.4 OpenDaylight SDN Controller

In 2013, the open network foundation with the Linux Foundation created the OpenDaylight (ODL) SDN controller. The motivational factor for the development of ODL was the prevalence of issues within previously released SDN controllers. As a result, a new SDN controller was established and involved the collaboration of multiple vendors creating a more efficient and stable SDN controller. This has made ODL one of the most utilized open-source SDN controllers and has had a significant influence on commercially available SDN solutions [1].

One such utilization of ODL was within a research project conducted in 2018 [6] and utilized the ODL controller within a small topology designed for the deployment within an SME and utilized the underlay and overlay fabrics. They argued the importance of introducing network orchestration and automation to promote scalability and simple operation. Within this study, they successfully deployed ODL as a viable SDN controller within a small academic or SME topology [6] and utilized various APIs available within ODL to facilitate network automation. This study and the collaborative approach to the development of ODL show this as a potentially



viable SDN controller within an SME network. However, the study failed to discuss how the underlay and overlay fabrics were introduced into the topology in which the 'NetO-App' was introduced. Furthermore, it is recognized there are many other SDN controllers available, its impact within the paradigm of SDN development and research and resultant improvements within this sector makes ODL a significant choice when considering the implementation of SDN but the deployment method would need to be explored.

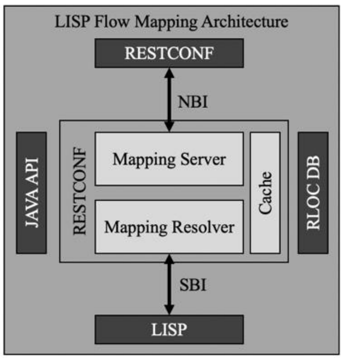
4 Deploying LISP, VxLAN within OpenDaylight SDN controller

For the deployment of an SDN-based underlay and overlay fabric utilizing LISP, VXLAN and an OpenDaylight SDN controller with smaller infrastructures, we show several examples in the form of prototype networks using the EVE-NG network virtualization tool to establish the viability of these technologies within an SME network.

4.1 Location Identifier Separation Protocol (LISP) within OpenDaylight

To facilitate LISP based activities within an SDN architecture, ODL provides the LISP Flow Mapping Services containing a Mapping Server (MS) and Mapping Resolver (MR) [19](Fig. 3). This architecture operates similarly to LISP within Cisco DNAC, this provides a centralized location where EIDs can be mapped to their respective RLOC. Similarly, the Mapping Resolver, analogous to the Domain Name Service (DNS), resolves requests and updates sent from RLOCs within the SDN network. This is achieved by using the LISP protocol through the SBI and allows the register and request of RLOCs from any device that supports this protocol. The RESTCONF API can be used to explicitly map EID to RLOC within the mapping server through the LISP Flow Mapping Service NBI [19] providing programmatic opportunities to introduce automation into EID and RLOC mapping. Whilst the LISP protocol is a Cisco proprietary technology, it is available as an open-source technology and could provide the basis of underlay fabrics within an open-source SDN solution such as OpenDaylight.

Fig. 4 shows LISP deployment within a network, we created a core consisting of five CSR1000v routers, of which three have been selected to operate as the Routing Locators (RLOC) and function as both ingress and egress routers (xTR) into the core network. To provide connectivity between each of the xTR devices, OSPF has been configured to provide routing within the network Core. Each of the campus routers has been introduced to mimic host devices connected to the inside interface of the xTR and example LISP over a geographical separation. Within the Core-MS-MR router, two roles are performed, the first is the Map-Server and allows each of the RLOC routers to register the connected devices' Endpoint Identifiers (EID) and form an RLOC-to-EID database within the core router.



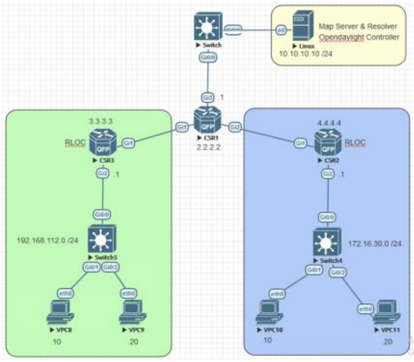


Fig. 3 ODL LISP Flow Mapping Architecture   **Fig. 4** LISP Network Diagram

The second role is the Map-Resolver which handles Map-Requests (MR) from xTR devices when the device requires the location of an EID within the network. The MR performs a lookup to match the destination EID with the RLOC it is connected to and informs the xTR which RLOC the data should be forwarded on to.

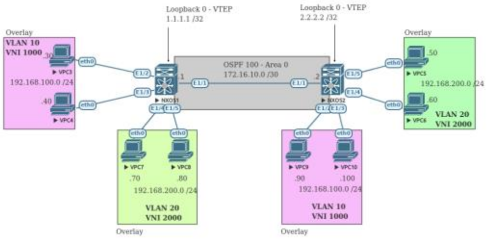
To demonstrate the viability of LISP deployment with a small network, we removed LISP services from the router and utilized it as a Core edge access router with an Ubuntu Server operating behind the Core edge device. Fig.5 shows the network diagram and deployment within EVE-NG virtualization software with CSR3 and CSR2 providing xTR operations. Similar to the previous demonstration, OSPF has been used for Core routing and the LISP Mapping services moved to a centralized OpenDaylight SDN Controller which deployed on an ubuntu server.

Fig. 5 LISP and OpenDaylight SDN Network topology

The key benefit of introducing the OpenDaylight SDN controller to services such as LISP is the programmable opportunities within the network. This enables organizations to create solutions and programs that a unique and fully tailored towards the organization's network requirements. Whilst controllers such as OpenDaylight can be utilized to operate and manage many networking aspects, Rest APIs within the controller can be used to abstract information which can be later used to make programmatic decisions and network changes.

4.2 Virtual Extensible Local Area Networks & OpenDaylight

As previously discussed with the Cisco DNAC-based literature, the deployment of underlay and overlay networks could be achieved through the implementation of VXLAN technologies. The VXLAN technology can provide layer 2 extensions of segmented networks across different domains and increasing the number of available LANs significantly compared to traditional VLAN segmentation [17]. In a study of virtual machine (VM) migration between distributed datacenters, the OpenDaylight SDN controller was successfully utilized to orchestrate the efficient migration of the



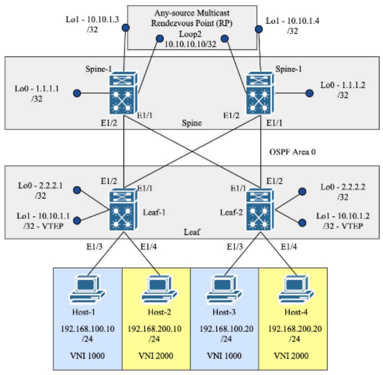
VMs both inter-domain and inter-LAN through the utilization of VXLAN. Applying this approach to network segmentation would allow both the migration and movement of physical devices and virtual devices.

Typically, VXLAN uses the 'Flood and Learn' approach to distribute broadcast, unknown unicast and multicast (BUM) traffic to every VTEP device within the same multicast group. This reduces the need to flood broadcast traffic throughout the entire network and limiting network device utilization. This can be achieved within the data plane using either multicast or the 'Ingress Replication' method. An advantage of using multicast VXLAN is the scalability through network growth, as long as the switch resides within the same multicast group and the VXLAN process, it will receive BUM traffic for the configured networks. As a solution, we have demonstrated how VXLAN can be implemented within a small network using the 'Flood and Learn' approach using firstly the unicast ingress replication method (Fig. 6) and secondly, the multicast method (Fig. 7).

Fig. 6 VXLAN unicast ingress replication

Fig. 7 VXLAN Any-Source Multicast

In some scenarios, data transfers may require windowing and error-checking mechanisms as found within TCP. As such, using a multicast VXLAN would be unsuitable as this method of messaging uses the less reliable but more efficient UDP transport protocol [16]. However, within a scenario where the IP address is known but the MAC address is not, then ARP requests would need to be broadcast throughout the Layer 2 network to establish the destination MAC. Since we are operating L2 over L3 overlay mechanisms need to be introduced to ensure replication of broadcast messages within the underlay. A solution is the manual replication of unknown unicast traffic through Protocol-Independent Multicast (PIM). Fig. 6 shows a network diagram of a small network where PIM is introduced to manually replicate (multicast) unknown unicast traffic and sent to pre-configured VTEP destinations. However, organizations using this method would not benefit from automatic device discovery as provided within Cisco DNAC. In response to this, the Any-source Multicast Rendezvous Point utilizing the 'Flood and Learn' concept was exampled to enable automatic discovery of VTEP devices within the same multicast group (Fig. 7). This method of VXLAN deployment provides small business scalability opportunities through expansion of layer 2 over layer 3 campus sites whilst simplifying the configuring new devices with network growth.



5 Conclusion

The key motivation of this study was to examine how SDN concepts could be introduced into small to medium organisations using open-source alternatives to minimize financial costs in comparison to commercially available systems. We examined some of the key features within both commercially available controllers and open-source alternatives with some features drawing similarities. Of significance is the professional and comprehensive solution provided by Cisco DNAC and its end-user interface. This would be a good option for organizations seeking to deploy SDN within their network but considering budgets, it may not be a financially viable option. However, through our research, several key features of Cisco DNAC such as LISP and VXLAN were matched with services that could be introduced using an open-source SDN controller such as OpenDaylight (ODL). The functionalities provided by the ODL SDN controller to support services such as LISP and VXLAN drawing significant similarities to the Cisco SD-Access architecture. This would allow for an open-source alternative to introduce underlay and overlay fabrics within SME infrastructure with zero-cost compared to Cisco DNAC. Organizations would still need to establish whether current networking hardware can support technologies such as LISP and VXLAN.

Although we have shown open-source alternatives to commercial solutions, further quantitative research would provide opportunities to analyze metrics such as performance and effectiveness between open source and commercial solutions.

References

1. Badotra, S., & Panda, S. N. (2020). Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking. Cluster Computing, 23(2), 1281–1291. https://doi.org/10.1007/s10586-019-02996-0
2. Bhutani, A., & Wadhwani, P. (2019). Software Defined Networking (SDN) Market Size by Component. https://www.gminsights.com/industry-analysis/software-defined-networking-sdn-market
3. Cisco Systems Inc. (2020). Cisco Validated Design Program. https://www.cisco.com/c/en/us/solutions/enterprise/validated-design-program/networking_solutions_products_genericcontent0900aecd80601e22.html
4. Cisco Systems. (2019b). Programmability Configuration Guide , Cisco IOS XE Amsterdam 17 . 1 . x. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/171/b_171_programmability_cg/restconf_protocol.html
5. Godanj, I., Nenadić, K., & Romić, K. (2016). Simple example of Software Defined Network. Proceedings of 2016 International Conference on Smart Systems and Technologies, SST 2016, 231–238. https://doi.org/10.1109/SST.2016.7765665
6. Gedia, D., & Perigo, L. (2018). Neto-App: A Network Orchestration Application for Centralized Network Management in Small Business Networks. IT in Industry, 6(3), 61–72.
7. Gooley, J., Schuemann, D., Yanch, D., Curran, J. (n.d.). Cisco Software-Defined Wide Area Networks: Designing, Deploying and Securing Your Next Generation WAN with Cisco SD-WAN. United Kingdom: Cisco Press.
8. Huang, T., Yu, F. R., Zhang, C., Liu, J., Zhang, J., & Liu, Y. (2017). A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. In IEEE

Communications Surveys and Tutorials (Vol. 19, Issue 2, pp. 891–917). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/COMST.2016.2630047

9. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., & Vahdat, A. (2013). B4: Experience with a globally-deployed software defined WAN. Computer Communication Review, 43(4), 3–14. https://doi.org/10.1145/2534169.2486019
10. Jarschel, M., Zinner, T., Hossfeld, T., Tran-Gia, P., & Kellerer, W. (2014). Interfaces, attributes, and use cases: A compass for SDN. IEEE Communications Magazine, 52(6), 210–217. https://doi.org/10.1109/MCOM.2014.6829966
11. Kim, H., Kim, J., & Ko, Y.-B. (2014). Developing a cost-effective OpenFlow testbed for small-scale Software Defined Networking. 16th International Conference on Advanced Communication Technology, 758–761. https://doi.org/10.1109/ICACT.2014.6779064
12. Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015a). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999
13. Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015b). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999
14. Levin, D., Schmid, S., Canini, M., Schaffert, F., & Feldmann, A. (2014). Panopticon: Reaping the Benefits of Incremental SDN Deployment in Enterprise Networks. 2014 USENIX Annual Technical Conferenc, 333.
15. Lynn, T., Mooney, J. G., Rosati, P., & Fox, G. (2020). Measuring the Business Value of Cloud Computing. 1–123. https://doi.org/10.1007/978-3-030-43198-3
16. Naranjo, E. F., & Salazar Ch, G. D. (2018). Underlay and overlay networks: The approach to solve addressing and segmentation problems in the new networking era: VXLAN encapsulation with Cisco and open-source networks. 2017 IEEE 2nd Ecuador Technical Chapters Meeting, ETCM 2017, 2017-Janua, 1–6.
17. Noghani, K. A., Kassler, A., & Gopannan, P. S. (2018). EVPN/SDN Assisted Live VM Migration between Geo-Distributed Data Centers. 2018 4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018, 182–186.
18. Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys and Tutorials, 16(3), 1617–1634.
19. OpenDaylight Project. (2018a). LISP Flow Mapping User Guide — LISP Flow Mapping master documentation.
20. Prajapati, A., Sakadasariya, A., & Patel, J. (2018). Software defined network: Future of networking. Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018, 1351–1354.
21. Rodriguez-Natal, A., et al., "LISP: a southbound SDN protocol?," in IEEE Communications Magazine, vol. 53, no. 7, pp. 201-207, July 2015, doi: 10.1109/MCOM.2015.7158286.
22. Tan, K. S., Chong, S. C., Lin, B., & Eze, U. C. (2010). Internet-based ICT adoption among SMEs: Demographic versus benefits, barriers, and adoption intention. Journal of Enterprise Information Management, 23(1), 27–55.
23. Tarutė, A., & Gatautis, R. (2014). ICT Impact on SMEs Performance. Procedia - Social and Behavioral Sciences, 110, 1218–1225. https://doi.org/10.1016/j.sbspro.2013.12.968
24. Zhao, Y., Iannone, L., & Riguidel, M. (2016). On the performance of SDN controllers: A reality check. 2015 IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015, 79–85.
25. Zhu, L., Karim, M. M., Sharif, K., Li, F., Du, X., & Guizani, M. (2019). SDN Controllers: Benchmarking & Performance Evaluation. ArXiv. http://arxiv.org/abs/1902.04491