ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 5, September 2015

Future Infrastructure-Less Wireless Networking Environments: A Software Defined Network Approach

Sneha Paruchuri
Department of Electrical Engineering, New York Institute of Technology, Old Westbury, NY

Abstract- ***The growing popularity of smart phones, tablet computers and cloud services places an increasing demand for dynamic services from wireless networks. This demand generates new requirements to tackle ever-increasing quality of experience requirements for the network architecture such as flexibility in management and configuration, adaptability and vendor-independence. To meet these requirements, a new architecture consisting of hardware running a multitude of specialized network software is required. This network software may be data-plane software providing network function virtualization (NFV), or control-plane software providing centralized network management - software defined networking (SDN). Although manufactures of wireless equipment are increasing their involvement in SDN-related activities, to date there is not a clear and comprehensive understanding of what are the opportunities offered by SDN in most common networking scenarios involving infrastructure-less wireless communications and how SDN concepts should be expanded to suit the characteristics of wireless and mobile communications. This paper is a first attempt to fill this gap. In fact, it aims at analyzing how SDN can be beneficial in infrastructure-less wireless networking environments, and how it should be expanded to take the characteristics of such networking environments into account. This paper describes the emergence of SDN as an important new networking technology. The main focus is to explore the salient features of SDN, the network architecture with SDN, opportunities of SDN, the benefits of SDN wireless SDN techniques, and SDN for infrastructure-heavy wireless networks.***

Keywords- **Software defined networking, Network architecture. Wireless SDN techniques, infrastructure-less wireless networking, benefits of SDN.**

I. INTRODUCTION

The saturation of our world with information processing capacity heralds a paradigm shift in computer applications tiny, cheap processors embedded into many everyday objects can detect their surroundings via similarly integrated sensors, and they can equip "their" object with both information processing and communications capabilities. This adds a completely new dimension to such objects they could, for example, find out where they are, what other objects are in their vicinity, and what had happened to them in the past. They may adapt to the environment, behave in a context-sensitive manner, and provide useful services in addition to their original purpose. They will be equipped with spontaneous network capabilities and will thus be able to communicate and cooperate with other smart objects and to access all sorts of Internet resources[1]. The risk profile related to data security, privacy and availability seeks a parallel, energy-efficient and high-security networking which is of utmost importance. Network operators and service and product providers require a new network solution to efficiently tackle the increasing demands of this changing network landscape. Software defined networking has emerged as an efficient network technology capable of supporting the dynamic nature of future network functions and intelligent applications while lowering operating costs through simplified hardware, software, and management.

SDN implementation opens up a means for new innovation and new applications. Dynamic topology control (i.e., adjusting switch usage depending on load and traffic mapping) becomes possible with the global network view. This introduces scope for network-wide access control, power management, and home networking, for which the network view is not beneficial but absolutely necessary. Furthermore, the network programmability possible in SDN allows seamless communication at all levels, from hardware to software and ultimately to end users (network operators).. However, despite their widespread adoption, traditional IP networks are complex and very hard to manage. The distributed control and transport network protocols running inside the routers and switches are the key technologies that allow information, in the form of digital packets, to travel around the world. Despite their widespread adoption, traditional IP networks are complex and hard to manage [2]. Automatic reconfiguration and response mechanisms are virtually non-existent in current IP networks. Enforcing the required policies in such a dynamic environment is therefore highly challenging. To make it even more complicated, current networks are also vertically integrated. The control plane (that decides how to handle

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 5, September 2015

network traffic) and the data plane (that forwards traffic according to the decisions made by the control plane) are bundled inside the networking devices, reducing flexibility and hindering innovation and evolution of the networking infrastructure.

II. SOFTWARE-DEFINED NETWORKING

SDN is an emerging networking paradigm that gives hope to change the limitations of current network infrastructures. [3], [4]. The Software-defined networking (SDN) is enabling organizations to simplify network configuration and to accelerate application deployment and delivery, dramatically reducing IT costs through policy-enabled workflow automation. SDN is envisioned as a way to reduce the complexity of network configuration and management while making the introduction of innovation in networks simpler [5]. SDN technology enables cloud architectures by delivering automated, on-demand application delivery and mobility at scale. SDN enhances the benefits of data center virtualization, increasing resource flexibility and utilization and reducing infrastructure costs and overhead. SDN accomplishes these business objectives by converging the management of network and application services into centralized, extensible orchestration platforms that can automate the provisioning and configuration of the entire infrastructure.

First, the forwarding abstraction should allow any forwarding behavior desired by the network application (the control program) while hiding details of the underlying hardware. The most notable example of such an abstraction is Open Flow [6], [7]. This abstraction breaks the vertical integration by separating the network's control logic (the control plane) from the underlying routers and switches that forward the traffic (the data plane). The data plane consists of the standard data packets that are forwarded by the network device. The logic to accomplish this forwarding is still at the hardware level. The control plane represents packets that manage the network devices. The control functionality has been lifted out of the hardware and is now defined in software, allowing network devices to be managed on-the-fly.

Second, the distribution abstraction should shield SDN applications from the vagaries of distributed state, making the distributed control problem a logically centralized one. Its realization requires a common distribution layer, which in SDN resides in the NOS. This layer has two essential functions. The first function is responsible for installing the control commands on the forwarding devices. With the separation of the control and data planes in the forwarding abstraction , network switches become simple forwarding device in a logically centralized controller simplifying policy enforcement and network (re) configuration and evolution [8]. The second function collects status information about the forwarding layer (network devices and links), to offer a global network view to network. This simple abstraction provides much needed flexibility to a variety of networking environments. The last abstraction is specification, which should allow a network application to express the desired network behavior without being responsible for implementing that behavior itself. This can be achieved through virtualization solutions, as well as network programming languages. It is important to emphasize that a logically centralized programmatic model does not postulate a physically centralized system [9]. Instead, production-level SDN network designs resort to physically distributed control planes [10]. The separation of the control plane and the data plane can be realized by means of a well-defined programming interface between the switches and the SDN controller.

These approaches map the abstract configurations that the applications express based on a simplified, abstract model of the network, into a physical configuration for the global network view exposed by the SDN controller.

A. Open Flow

Open Flow is the most notable instantiation of Software-Defined Networking principles, via the decoupling of the data and control planes. Open Flow's architecture requires forwarding devices, also referred to as Open Flow switches, and a network controller The controller propagates forwarding rules into each switch's flow table, a construct that contains information necessary to maintain the Open Flow specification [11]. This allows the forwarding devices to make decisions based on flows of traffic, a level of control that was difficult to achieve before the days of Open Flow. An Open Flow switch has one or more tables of packet-handling rules. Each rule matches a subset of the traffic and performs certain actions (dropping, forwarding, modifying, etc.) on the traffic. Depending on the rules installed by a controller application, an Open Flow switch can – instructed by the controller – behave like a router, switch, firewall, or perform other roles (e.g., load balancer, traffic shaper, and in general those of a middle box). Though they technically are not a part of Open Flow, NOX and Flow Visor

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 5, September 2015

are tools that are very commonly used with Open Flow and are important to note. NOX is a network operating system that roughly acts as the network controller in the Open Flow framework. A NOX controller has a global view of the network and allows management and control applications to be run with ease. FlowVisor is a special controller that allows for virtualization in networks. Flow Visor easily achieves traffic isolation and other nice properties that were quite difficult to achieve before SDN

B. Network Architecture with SDN

- The control and data planes are decoupled. Control functionality is removed from network devices that will become simple (packet) forwarding elements.
- Forwarding decisions are flow-based, instead of destination-based. A flow is broadly defined by a set of packet field values acting as a match (filter) criterion and a set of actions (instructions). In the SDN/Open Flow context, a flow is a sequence of packets between a source and a destination. All packets of a flow receive identical service policies at the forwarding devices. The flow abstraction allows unifying the behavior of different types of network devices, including routers, switches, firewalls, and middle boxes. Flow programming enables unprecedented flexibility, limited only to the capabilities of the implemented flow tables .
- Control logic is moved to an external entity, the so called SDN controller or Network Operating System (NOS). The NOS is a software platform that runs on commodity server technology and provides the essential resources and abstractions to facilitate the programming of forwarding devices based on a logically centralized, abstract network view. Its purpose is therefore similar to that of a traditional operating system.
- The network is programmable through software applications running on top of the NOS that interacts with the underlying data plane devices. This is a fundamental characteristic of SDN, considered as its main value proposition

C. Opportunities of SDN

SDN is about simplification and evolvability. New network control and management solutions can be easily deployed on existing equipment, ideally, as simply as it is to install new programs on a computer. SDN paradigm envisions that the functionality performed at the network and higher layers of the protocol stack are defined through software and can be changed easily and "on the fly". In SDN there are network nodes (SDN switches) that are responsible for classifying packets in different flows and performing the corresponding actions. Packet classification is performed on the basis of certain rules. SDN switches can be distinguished from other nodes (the Controllers) that are responsible for setting the rules and corresponding actions. The extension of the SDN paradigm to wireless infrastructure-less networks, which we call Software Defined Wireless Networking (SDWN), can have significant advantages in today's networking environments.

D. The Benefits of SDN

Offering a centralized, programmable network that can dynamically provision so as to address the changing needs of businesses, SDN also provides the following benefits:

- Service provisioning speed and agility: Setting up networks in an SDN can be as easy as creating VM instances, and the way SDNs can be set up is a far better complement to VMs than plain old physical networks. SDN helps organizations rapidly deploy new applications, services, and infrastructure to quickly meet changing business goals and objectives.
- Network flexibility and holistic management: SDNs enable "network experimentation without impact"--meaning one can leap over the limits imposed by SNMP and experiment freely with new network configurations without being hamstrung by their consequences.
- Enable Innovation: SDN enables organizations to create new types of applications, services, and business models that can offer new revenue streams and more value from the network
- Better and more granular security: VMs have made network security a headache and a half. SDNs can provide the kind of fine-grained security for apps, endpoints and BYOD devices that a conventional hard-wired network can't.
- Efficiency and lower operating expenses: The exact cost savings of SDNs is still in doubt--for example, it's unclear whether it might simply shift costs to controllers and software. Still, 50% of the administrators surveyed who use SDNs said they sold the technology to their business executives as a money-saving methodology. And while many of those polled see lower hardware costs as a big SDN

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 5, September 2015

selling point, the bigger opportunity is lower opex costs due to improved network management efficiency, according to the report.

- Virtual network services, lowered capex: Even if the biggest benefits for SDNs will be in big-league data centers, there's still plenty of ways for enterprises to lower their capex --both by making better use of what enterprises already have, and by lessening dependencies on proprietary hardware and dedicated appliances. SDN potentially limits the need to purchase purpose-built, ASIC-based networking hardware, and instead supports pay-as-you-grow models
- Reduce OpEX: SDN enables algorithmic control of the network of network elements (such as hardware or software switches / routers that are increasingly programmable, making it easier to design, deploy, manage, and scale networks. The ability to automate provisioning and orchestration optimizes service availability and reliability by reducing overall management time and the chance for human error

E. Features of SDN

SDN has several features which contribute towards the enhancement and efficiency of networking, which makes it a promising future technology for networks and big data centers. Some of the prominent features [9] can be discussed as:

- Centralized Control: Network intelligence is logically centralized in SDN control plane, which gives us a global view of the network and the whole network appears as a single logical switch to applications. With SDN, network administrators gain vendor independent control over the entire network from a single logical point, which highly simplifies the network operations and design. Centralized control is very beneficial in case of distributed networking scenario.
- Abstraction and Virtualization: SDN uses abstract forwarding on each layer of layered approach used in SDN architecture, which hide the complexity of traffic flow in the network. SDN hides complexity of network from applications by providing logical view of network resources available and abstracting the actual traffic-flow control logic. Router can be divided into different virtual networks to implement different program logic on those networks.
- Programmability: SDN gives us freedom to write immediate program logic for controlling the data-flow dynamically. Instead in traditional networking, network devices like switches, routers etc., compute the best path for traffic flow by its own. Hence it increases the speed of data flow by minimizing the delay of path computation inside network devices as network devices only perform packet forwarding.
- Rapid Innovation: SDN helps in rapid innovation of new services deployment. In current traditional networking devices, services are already embedded with the hardware. These devices perform operations like path computation etc. by itself. So deployment of new application is limited to the services that came embedded with the hardware. However in SDN, control plane and data plane separation allows us to rapid deployment of unlimited services as networking devices only perform packet forwarding.
- Openness: SDN provides open standards due to which several open source communities like Open Flow, Open Networking Foundation, ON Lab etc. are working dedicatedly towards SDN. SDN contains open programming API's where any network administrator can write the control-logic of traffic flow according to its own infrastructure needs. Such openness allows flexibility and faster growth of new networking techniques.

III. WIRELESS SDN TECHNIQUES

The current technology involves deployment of a technique that has various drawbacks like Complexity, Inconsistent policies, Inability to scale and Vendor dependence. These drawbacks could be overcome using a new technology called Software Defined Networking. This technique involves deployment of all the above mentioned traditional techniques into software that manages everything in a network. SDN is a step in the evolution towards programmable and active networking. SDN allows network administrators to have programmable central control of network traffic via a controller without requiring physical access to the network switches. A set of open commands for forwarding was defined in the form of a protocol known as Open Flow. The OpenFlow protocol enables globally-aware software controllers. It is a prototype where a central software program called a controller, prescriptions the overall network behavior. In software defined networking network devices become simple packet accelerating devices (data plane) but the "common sense " or control logic is implemented in the controller (control plane). This exemplar shift brings several benefits compared to legacy methods. Techniques that permeate the different applications of SDN to wireless networks are as follows:

A. Slicing
Flow Visor is one of the early technologies to virtualize a SDN. Its basic idea is to allow multiple logical networks share the same Open Flow networking infrastructure. For this purpose, it provides an abstraction layer that makes it easier to slice a data plane based on off-the-shelf Open Flow-enabled switches, allowing multiple and diverse networks to co-exist. Five slicing dimensions are considered in Flow Visor: bandwidth, topology, traffic, device CPU and forwarding tables. Moreover, each network slices supports a controller, i.e., multiple controllers can co-exist on top of the same physical network infrastructure. Each controller is allowed to act only on its own network slice. In general terms, a slice is defined as a particular set of flows on the data plane. From a system design perspective, FlowVisor is a transparent proxy that intercepts OpenFlow messages between switches and controllers. It partitions the link bandwidth and flow tables of each switch. Each slice receives a minimum data rate and each guest controller gets its own virtual flow table in the switches.

B. Control Strategies
As mentioned in the previous section, Open Flow can have one centralized controller or many distributed controllers. There are positives and negatives to each approach. With a centralized controller, global network state only needs to be maintained on one controller. As a result, no controller-to-controller communication strategy is necessary. In the Open Flow architecture, there is no standardized communication protocol between controllers, so this is a reasonable option. Having one controller is also the cheapest option, which makes it a more feasible approach for network operators with less capital. However, if the controller fails, there is no other controller available; unmatched flows sent to the failed controller are essentially dropped. If this controller performs any path management for the network, the paths at the time of failure will remain until the controller comes back online. This is particularly important in the wireless setting, which has much more variability than the wired networks Open Flow was designed for.

C. Traffic Engineering
The main goal of these applications is to engineer traffic with the aim of minimizing power consumption, maximizing aggregate network utilization, providing optimized load balancing, and other generic traffic optimization techniques. The most common application of traffic engineering in SDN is load balancing. Gathering data from the counters in the forwarding devices, the controller can form statistics to gauge how heavily the device is utilized. If this meets a certain threshold, it is intuitive that the controller could propagate a new rule to relieve that link's load, distributing the excess load to other routes. SDN load-balancing also simplifies the placement of network services in the network]. Every time a new server is installed, the load-balancing service can take the appropriate actions to seamlessly distribute the traffic among the available servers, taking into consideration both the network load and the available computing capacity of the respective servers. This simplifies network management and provides more flexibility to network operators.

It is intuitive that traffic engineering approach could extend to support energy-aware routing [12]. By measuring the energy output rather than the device load, the controller could easily route around network devices that could be sleeping. Some wireless network operators employ energy aware routing when the network utilization is not very high, but when the utilization increases they switch to a load balancing strategy. Other applications that perform routing and traffic engineering include application-aware networking for video and data streaming and improved QoS by employing multiple packet schedulers and other technique. As traffic engineering is a crucial issue in all kinds of networks, upcoming methods, techniques and innovations can be expected in the context of SDNs.

IV. SDN'S APPLICATION TO INFRASTRUCTURE-BASED WIRELESS NETWORK
Wireless networks have become so ubiquitous, they have a wide user base that wants high speeds for consuming increasingly-rich web content. This problem is industry wide. Cellular networks are tending towards smaller cells to find faster data rates, but smaller cells increase interference [13]. The typical user of a Wireless Local Area Network is a non-expert, so the ubiquity of these wireless networks has further increased chances for signal interference. Even ignoring interference with the signal, this consistent push for more speed often decreases the transmission range of the signal [14]. Similarly, more wireless networks are becoming

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 5, September 2015

oversaturated with devices. As a result, network operators are tending towards larger channel sizes to increase network capacity. However, this can also lower the transmission range. As these transmission ranges shorten, networks need a dense deployment of access points (APs) to cope with these problems, leading to more expenses. Wireless networks with dense deployments of access points are called infrastructure-heavy wireless networks. Intuitively, wireless networks that are light on infrastructure would need other strategies to provide connectivity to users.

A. SDN for Infrastructure-Heavy Wireless Networks

Wireless Local Area Networks (Wlans): Today's IEEE 802.11 enterprise WLAN deployments range from a few dozens to thousands of access points (APs), which need to serve a large number of users. These users connect to the enterprise WLAN through a multitude of devices, including smart phones, laptops, and tablets. Regardless of their size, these networks need to provide a varied set of services in a scalable manner. These services include support for authentication, access, and accounting (AAA), policy based network management, interference management, mobility management, dynamic channel reconfigurations, load balancing, intrusion detection and prevention, and providing quality of service guarantees. Many vendor solutions exist that cover a set of these features. However, these solutions are usually proprietary, and are closely tied to the hardware provided by the same vendor as well. There is thus a need for an open and flexible software architecture for enterprise WLANs.

In SDN, the network control plane is decoupled from the physical network topology and, instead, uses software to control how traffic is forwarded in the network. For instance, a switch's forwarding tables can be controlled remotely through a software controller (or network operating system). The Open Flow [6] protocol is considered an enabler of SDN because it provides a standardized protocol that can be used by a controller to manipulate forwarding tables of a network of switches (an analogy would be the x86 instruction set for computer architectures). One can now write network applications that can programmatically control the forwarding behavior of a network by talking to a network controller. Any Open Flow enabled switch from any vendor will have a common interface for forwarding plane manipulation via a controller (of which there are many open source implementations today). This enables flexible and simplified network management.

B. Cellular Networks

The application of SDN to mobile networks has unearthed scalability issues that were not encountered before. Particularly, mobile networks provide for a large number of subscribers who are frequently mobile, and the operators need to measure and control their traffic to provide services consistent with their business goals. The most fundamental advantage of applying SDN to mobile networks is the ability to distribute the data plane over multiple, cheaper network switches, as opposed to the packet gateways (P-GW) in the network. Previously, all traffic was required to go through P-GW. These P-GWs are extremely expensive (on the order of millions of dollars), so network operators frequently overloaded them. In other words, the operators would not buy as many P-GWs as they needed to save money, but this degraded the service quality of the network. However, with SDN operators can now buy a large number of cheaper devices from different vendors and distribute them, allowing for better traffic engineering with an SDN architecture, it would not be difficult to perform the measurements needed to properly bill the guest's mobile network operators. Lastly, because the controllers would have a global view of their base stations, they could be used for fine-grained subcarrier coordination to reduce inter-cell interference. SDN provides flexibility in a variety of infrastructure-heavy wireless network settings. In the infrastructure-heavy setting, there is much more of an industry bend. As a result, SDN's flexibility allows these dense wireless network operators to have a variety of vendor equipment, improve the network's latency, and perform cheap handovers between different wireless networking technologies.

V. SDN FOR LIGHT INFRASTRUCTURE WIRELESS NETWORKS

A. Wireless Mesh Networks

Wireless mesh networks (WMNs) are a cost-effective way to expand the coverage area of a WLAN [12]. A typical wireless mesh network will have a small number of nodes connected to the Internet; the rest of the nodes are connected to one of these nodes through a multi-hop path in the mesh topology. The topology allows for reduced outside connectivity, which is how WMNs save money. SDN is particularly applicable to WMNs because of the routing decisions necessary for inner nodes to communicate with the Internet. Broadly, SDN provides a global view at the controller, and per-flow routing capabilities to allow for traffic engineering and reliability. It is intuitive that these traits are important in a WMN. OpenFlow and NOX provided the flow-based management as desired.

B. Home Networks

Applying SDN to home wireless networks is a relatively new pursuit. There is a fundamental difference between the tools of the Internet, which were largely designed with enterprises in mind, and the tools necessary for non-experts to properly manage their home networks [13]. Home networks are becoming increasingly heterogeneous, as home users typically connect many different devices to one wireless router. This is particularly because devices have very different use cases, such as entertainment, work, and communication [16]. There are two ways SDN is applied to home networks [15] [16], and they both take a similar approach, seeking to provide simpler management, per-flow control, traffic isolation, and a more intuitive management interface. Both groups use NOX controllers and OpenFlow as the southbound protocol. However, there are differences, particularly in how they achieve traffic isolation and the provided management schemes. Used custom DHCP and DNS implementations in past to provide fine-grained control of the home network. Specifically, they used DHCP to control the associations to their network, which naturally extended itself to enforce temporary leases and guest

C. Rural Wireless Network Operators

Perhaps the most challenging application of Software-Defined Networking is wireless Internet service providers (ISPs) in emerging markets, as they are markedly resource constrained. In [17], the authors label these constrained wireless ISPs with the term rural wireless network operators (RWNOs). RWNOs must scrape together a working wireless network from very different devices, with the hope that the network can withstand the harsh environment around it, while still delivering acceptable service.

VI. CONCLUSION

SDN has emerged as a means to improve programmability within the network to support the dynamic nature of future network functions. As bandwidth demand escalates, the provision of additional capabilities and processing power with support for multiple 100GE channels will be seamless through an SDN-based update and/or upgrade. SDN promises flexibility, centralized control, and open interfaces between nodes, enabling an efficient, adaptive network. In order to achieve this goal, a number of outstanding challenges must be resolved. In this article we have presented a discussion of a number of challenges in the area of performance, scalability, security, and interoperability. Openness of SDN system allows people to write control programs so it is essential to design some protocols or use existing protocols efficiently that will check the correctness of programming logic before implementation i.e. it will check the authentication and authorization so as to prevent the collisions of data inside the network which causes congestion. Given the global view, consistency of policies is straightforward to enforce. SDN has successfully managed to pave the way towards a next generation networking, spawning an innovative research and development environment, promoting advances in several areas: switch and controller platform design, evolution of scalability and performance of devices and architectures, promotion of security and dependability. We will continue to witness extensive activity around SDN in the near future. Emerging topics requiring further research are, for example: the migration path to SDN, extending SDN towards carrier transport networks, realization of the network as-a-service cloud computing paradigm, or software-defined environments (SDE).

REFERENCES

[1] Javed Mohammed, "Evolution of the Next Generation of Technologies: Mobile and Ubiquitous Computing", International Journal of Advanced Research in Science, Engineering and Technology Vol. 1, Issue 5 , December 2014, ISSN: 2350-0328.

[2] T. Benson, A. Akella, and D. Maltz, “Unraveling the complexity of network management,” in Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, ser. NSDI’09, Berkeley, CA, USA, 2009, pp. 335–348.

[3] N. Mckeown, “How SDN will Shape Networking,” October 2011. [Online]. Available: http://www.youtube.com/watch? v=c9-K5OqYgA.

[4] S. Shanker, “The Future of Networking, and the Past of Protocols,” October 2011. [Online]. Available: http://www.youtube.com/watch?v= YHeyuD89n1Y.

[5] S. Shanker et al.. "The future of networking and the past of protocols". OpenNetSummit 2011. October 2011.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shanker, and J. Turner, "Open Flow: enabling innovation in campus networks," SIGCOMM Computer Communication Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008.

[7] ONF, "Open networking foundation," 2014. [Online]. Available: https://www.opennetworking.org/.

[8] H. Kim and N. Feamster, "Improving network management with software defined networking," Communications Magazine, IEEE, vol. 51, no. 2, pp. 114–119, 2013.

[9] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: a distributed control platform for large-scale production networks," in Proceedings of the 9th USENIX conference on Operating systems design and implementation, ser. OSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–6.

[10] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. H¨olzle, S. Stuart, and A. Vahdat, "B4: experience with a globally-deployed software defined wan," in Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 3–14.

[11] Nick McKeown et al., "Open Flow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, Vol. 38, Issue 2, pp. 69-74, 2008.http://archive.openflow.org/documents/openflow-wp latest.pdf.

[12] Salvatore Costanzo et al., "Software Defined Wireless Networks: Unbridling SDNs," European Workshop on Software Defined Networking, pp. 1-6, 2012 .http://ieeexplore.ieee.org/xpl/articleDetails.jsp? a number=6385039.

[13] Christos Kolias et al., "OpenFlow-Enabled Mobile and Wireless Networks," Open Networking Foundation Solution Brief, pp. 1-13, 2013.https://www.opennetworking.org/solution-brief-openflow-enabled-mobile-and-wireless-networks.

[14] Peter Dely, "Architectures and Algorithms for Future Wireless Local Area Networks," Doctoral Thesis, Karlstad University, pp. 1-256, 2013. http://www.diva-portal.org/smash/get/diva2:563553/FULLTEXT01.pdf.

[15] R. Mortier et al., "Control and Understanding: Owning Your Home Network," Fourth International Conference on Communication Systems and Networks, pp. 1-10, 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6151322.

[16] Yiannis Yiakoumis et al., "Slicing Home Networks," Proceedings of the 2nd ACM SIGCOMM Workshop on Home Networks, pp. 1-6, 2011.http://yuba.stanford.edu/~nickm/papers/Homenets2011.pdf.

[17] Shaddi Hasan et al., "Enabling Rural Connectivity with SDN," University of California, Berkeley Technical Report, 2012.http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-201.html.

AUTHOR BIOGRAPHY

Sneha Paruchuri **was born in Vijayawada**, Andhra Pradesh, India, in 1991. She received the Bachelor's in Engineering in Electrical and Computer Engineering from the Acharya Nagarjuna University, India, in 2013. She did her Masters in Electrical and Computer Engineering from **NewY**ork Institute of Technology and graduated in May 2014. She is planning to pursue PhD in wireless sensors and networking offered through the new Cylab Mobility Research Center located at the Carnegie Mellon Silicon Valley campus. From 2011, she worked on various multi-disciplinary projects in wireless sensor network (WSN), privacy-preserving data aggregation for WSNs, and mobile cloud computing (MCC) projects as well as on Java and Oracle ODI projects. Her current research interests include wireless sensor networks, cryptanalysis data confidentiality in cloud-assisted wireless body area networks, Mobile and Wireless Enterprises, Oracle ODI, and Java. Ms. Sneha Paruchuri is a member of IEEE (Institute of Electrical and Electronics Engineers, Inc), Society of Women Engineers, Envi-Gis Group, Big Data Group, and also a volunteer of STEM program and REACH HIGHER program (Initiative of the First Lady Michelle Obama) at the NYIT, Old Westbury Campus. Ms Paruchuri was instrumental in empowering STEM education through various research articles one of which she submitted a paper titled" Next Generation Innovative Initiatives and Technologies for STEM Education within Asian American and Pacific Islanders(AAPI) Community" to AAPI community(4-14-2014) at the U.S. Department of the Interior **.**