Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 7 No. 4, December, 2016 - www.cisdijournal.net

Software Defined Networking (SDN) Application for Bring-Your- Own-Device (BYOD) Implementation in Library Automation

Yusuf, A.A.
Information and Communication Technology Department
Federal University of Petroleum Recourses
Effurun, Delta State, Nigeria
E-mail: waheebol@yahoo.com
Phone: +2348058740228

ABSTRACT

Library Automation facilitate speedy access to library resources within the shortest possible time through a computer network, minimizing the necessity of human intervention in operation. However, its benefits could not be fully optimised due to challenges posed by paucity of fund and incompatibility of hardware required to access such network resources. To mitigate this challenge, this paper focused on the application of Software-Defined Networking (SDN), a new paradigm in networking that facilitate a programmable network, for implementing Bring-Your-Own-Devices (BYOD) to access the automated library resources on the network. The design approach adopted for the proposed model takes into consideration the Security concern for BYOD Implementation.

Keywords: Library Automation, SDN, BYOD, OpenFlow, Programmable Network & Network Resources

CISDI Journal Reference Format
Yusuf, A.A. (2016): Software Defined Networking (SDN) Application for Bring-Your- Own-Device (BYOD) Implementation in Library Automation. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 7 No 4. Pp 165-170.
Available online at www.cisdijournal.net

1. INRODUCTION

The main aim of any library is to provide access to proper information, to the right users in as possible as short time. In an environment of information explosion, due to growing demands of the user and shrinking of financial resources, library is not able to obtain all the reading materials on demand. The only way to overcome these problem is resources sharing thorough networking (Shahaji & Shri, n.d.). Therefore, it is important to automate library manual processes on a computer network with specialised software that guarantee speedy access to library information. Library automation may be defined as the application of automatic and semiautomatic data processing machines (computers) to perform traditional library housekeeping activities such as acquisition, circulation, cataloguing and reference and serials control. Today "Library Automation" is by far the most commonly used terms to describe the mechanization of library activities using the computer (Uddin, 2009).

Now a day Library Automation has become the buzz word in library profession and has become a bare necessity for any libraries. An automated library can provide better library services to their users and can maintain the library more properly which a manual library can't do. The record keeping activities and various report generation becomes very easy in an automated library system (Debasis & Parnab, 2015). Though the benefits of Library Automation are enormous. Rajput & Gautam (2010) identified paucity of fund and incompatibility of hardware as some of the challenges that limit its implementation. Implementing Bring Your Own Device (BYOD) for client access in Library Automation will address these challenges to a great extent, however, security threats posed by its implementation remain a great concern.

2. RELATED LITERATURE

Professional Development Service for Teachers (PDST) Technology in Education (n.d.) while describing the concept of BYOD submits: *"Bring your own device (BYOD) involves allowing pupils/students to bring their own devices, especially tablets and other suitable personal devices, into classrooms to support improving student learning outcomes. Rather than the school owning the computing devices, as has been the norm to date, the devices are student owned. BYOD can be used both as an alternative to, or as an additional level of support to computers owned and provided by the school, such as a set of tablets that are being shared between classrooms at different times. Many BYOD suitable devices may be more capable and up to date than some school computers. Schools are beginning to see the potential of BYOD to support a more student-centered, active learning approach, with students taking more responsibility for their own learning."*

However, malware infected clients, denial of service and similar attacks are capable of bringing down the whole network. The situation becomes further compounded as identifying user-own devices is not easy, and thus difficult to manage by the IT department.

2.1 Previous Implementations of BYOD

In order to overcome the challenges posed by implementing BYOD, Android 2.2 introduces support for enterprise applications by offering the Android Device Administration (ADA) API. The Device Administration API provides device administration features at the system level. These APIs allow you to create security-aware applications that are useful in enterprise settings, in which IT professionals require rich control over employee devices such as strong password policy employment and remote device wiping. Users will need to activate the admin application on their android devices before they can get any of the application benefits available on the network. However, there is room for an android device to have more than one admin application actively running on it, where the one with the strictest policy is enforced (Device Administration, n.d.)

Miradore Ltd. (2016) describes Miradore Online that provides remote control and management of devices by installing a client version on the device. The client is responsible for running the management tasks in the managed device like gathering inventory, administrating package installations, and communicating with the Miradore server. It does not provide a granular control over the network

Samsung (2016) describes Knox, a device/app data control and protection that separate, isolate, encrypt, and protect work data on mobile devices. The work data can be manage remotely by the company while personal information such as pictures and messages remain private. This lack the adequate management features required to actively mitigate security threats such as access control.

2.2 Software Defined Networking (SDN)

Computer networks are typically built from a large number of network devices such as routers, switches and numerous types of middleboxes (i.e., devices that manipulate traffic for purposes other than packet forwarding, such as a firewall) with many complex protocols implemented on them. Network operators are responsible for configuring policies to respond to a wide range of network events and applications. They have to manually transform these high level-policies into low-level configuration commands while adapting to changing network conditions. And often they need to accomplish these very complex tasks with access to very limited tools. As a result, network management and performance tuning is quite challenging and thus error-prone. The fact that network devices are usually vertically-integrated black boxes exacerbates the challenge network operators and administrators face (Bruno et al, 2014).

Software-Defined Networking (SDN) enables innovation within the network by allowing networking equipment behavior to be modified. Forwarding decisions within a Software-Defined Network are made by software residing external to the switches; the switches merely forward traffic based on the decisions made by the control software. New protocols can be tested and deployed, and the network can be customized to particular applications, by replacing only the control software. Upgrading or replacing software is far easier than upgrading or replacing hardware (Glen, 2013). The Open Networking Foundation (2016), defines SDN as: *"The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.''*

Accordingly, SDN is viewed as a new paradigm in networking that focuses on the creation of programmable network by decoupling the data plane (forwarding Hardware) from the control plane, and employs a central software-based controller for network orchestration and management. These forwarding hardware at the data plane can be programmed to meet varying organisational needs via an open interface such as Open Flow.

The open, programmable interfaces allow for flexible interactions between the networking applications and the underlying physical network (i.e., the data plane) that is employed to provide networking services to the applications. In particular, the OpenFlow (OF) protocol provides a standardized interface between the control plane and the underlying physical network (data plane) (Andreas et al, 2015). Consequently, Open flow protocols handles the communication between the controller and the forwarding hardware (Southbound) based on flow tables created by matching defined rules, actions/instructions to be executed when the flow matches the rules and counters for collecting flow statistics

Software-Defined Networking was developed to facilitate innovation and enable simple programmatic control of the network data-path. As visualized in Figure 1, the separation of the forwarding hardware from the control logic allows easier deployment of new protocols and applications, straightforward network visualization and management, and consolidation of various middleboxes into software control. Instead of enforcing policies and running protocols on a convolution of scattered devices, the network is reduced to "simple" forwarding hardware and the decision-making network controller(s) (Bruno et al, 2014).

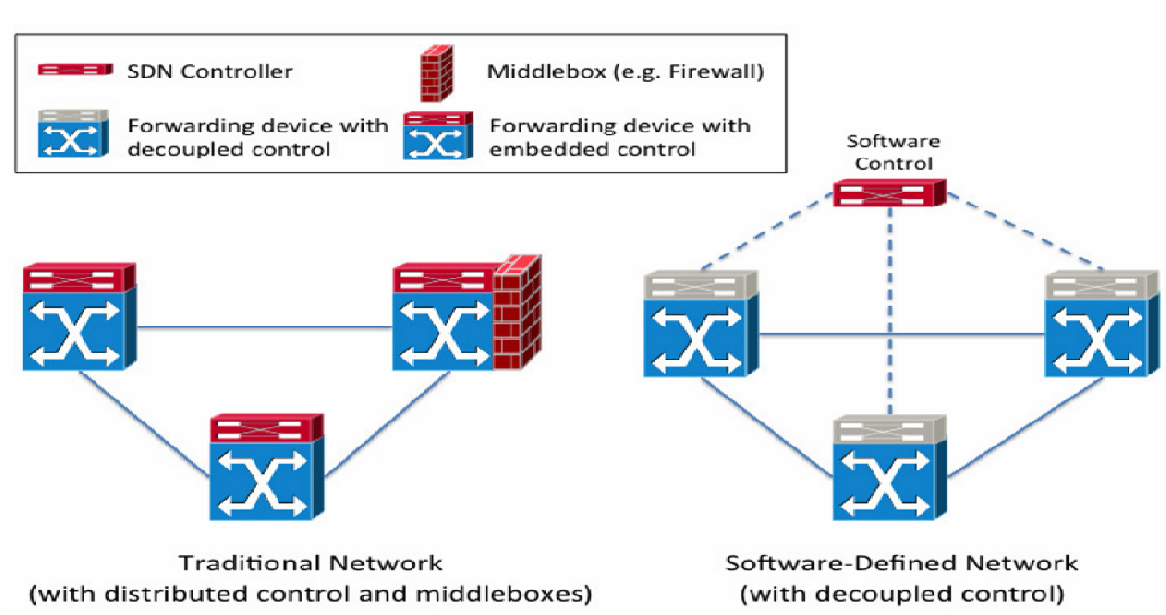

Fig. 1. The SDN architecture decouples control logic from the forwarding hardware, and enables the consolidation of middleboxes, simpler policy management, and new functionalities. The solid lines define the data-plane links and the dashed lines the control-plane links (Bruno et al, 2014).

.

3. STATEMENT OF THE PROBLEM

Some of the major challenges faced by Library Automation are paucity of fund and incompatibility of hardware required to access such network resources. These has greatly hinder its implementation, limiting access to library resources.

3.1 Research Justification

In order to eradicate these challenges, this research project focus on the implementation of BYOD on SDN model that connects clients of library automation using commercial wireless routers/Access Points and physical switch with hardware features of Ethernet Switch ASIC's (Application Specific Integrated Circuits), and proactively mitigate the security threats associated with BYOD operation on a network.

3.2 Research Direction

The objective of this research project is to design an SDN (OpenFlow) based programmable network model with the following features:

1. Employs commercial wireless routers/access points and swicthes with ethernet ASIC's features by loading some special software to support OpenFlow.
2. Employs Access Control to proactively mitigate security threats associated with network that conveys library automation.
3. A Bandwidth Management System that prevent infected BYOD clients from generating excessive traffic that can hamper the speed of the network.
4. Supports single login for network connectivity and access to library resources.
5. An automated system that mirrored network configurations to a backup server for network availability.

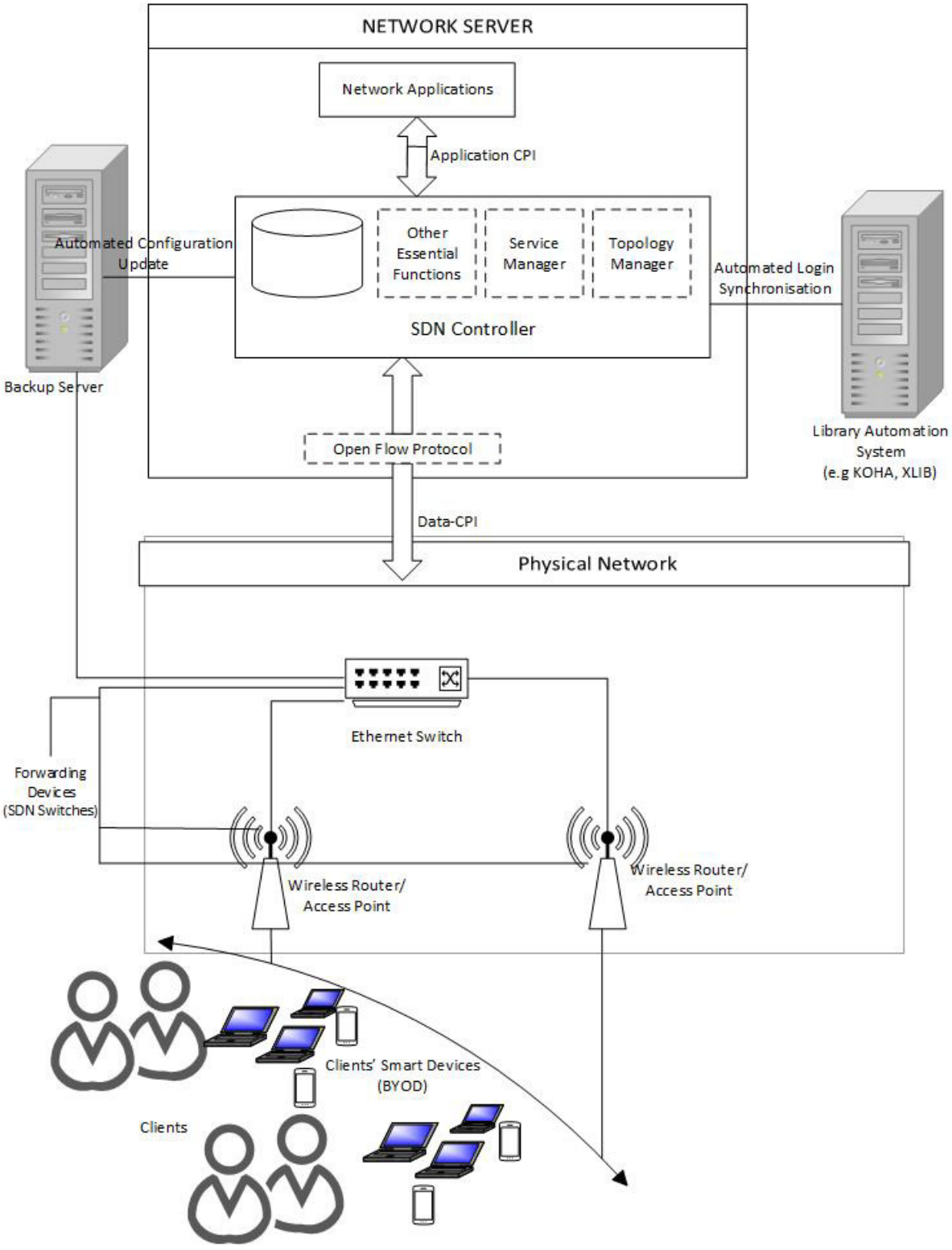

Fig. 2. The Architecture of the proposed SDN Model

4. METHODOLOGY

Figure 2 shows the architectural diagram for the proposed SDN Model. The SDN (Open Flow) controller is viewed as a network operating system that provides a programmatic interface for the network that can be used for implementing Management/orchestration tasks and offer new functionalities. The controller interfaces with the Network Applications through North-Bound Programmers Interface or Application Control Plane Interface (Application CPI), by providing an abstraction layer that virtualise the physical network to network application, with the assumption of centralized control where each network application such as email (Simple Mail Transfer Protocol) and web (Hyper Text Transfer Protocol) views the entire SDN Network as a single entity rather than a collection of individual devices. Other High level network applications that implement security policies are implemented at the interface. Table 1 give the details of security policies and the corresponding configurations. Theses configurations are manually made by the Network Administrators at the controller.

The controller interface with the data plane through South-bound Application Programmers Interface or Data Control Plane Interface (D-CPI) using the open flow protocol. It forward traffic base on flow table created by matching node addresses (IP or MAC addresses) with defined rules and policies, as may be communicated by the controller. The choice of controller adopted is Maestro. Maestro keeps a simple single-threaded programming model for application programmers of the system, yet enables and manages parallelism as a service to application programmers. It exploits parallelism in every corner together with additional throughput optimization techniques to scale the throughput of the system (Zheng et al, 2010)]. Accordingly, the controller proactively installs flow entries on the switches as defined in the security policies. The switches receives these flow entries as updates from the controller, and statically keeps such entries. In an instance where a switch receives a packet from a host through an inport, it will match the packet source address against the conditions stipulated by the rules in the flow entries. If the conditions are met, the packet will be forwarded to the destination through the outport of the switch, else the packet will be dropped. The switch do not need to communicate with the controller every time packet is received as it can forward traffic base on its flow entries, however, communication is established between the switch and controller whenever there is need for new flow entry.

Network Provisioning is done by the service manager subcomponent of the controller. It handles network task such as Domain Name services, dynamic address allocation to host and so on, while the topology manager manages the abstraction of the network attribute of the underlying physical network. The controller has an integrated database that is used to grant access to the clients' devices via a wireless connection by matching the supplied login parameters to the parameters contained in the database. These parameters consist of user id (matric number/staff id) and passwords. The database is updated manually by an administrator at the back end. The open interface allows the programming of the controller to perform additional essential functions such as automatically updating (mirroring) network configurations to the backup server; and login synchronisation to the Library Automation System. The former process ensures minimum network downtime in case of Network Server failure (Network Availability), while the later allows the clients to connect to the network and access the library resources with a single user account. Dynamic Host Control Protocol (DHCP) automatically allocate internet protocol parameters such as IP addresses and default gateway to BYOD clients as it connect to the SDN network through the wireless media before supplying the login parameters for authentication via a web browser such as Google chrome. The SDN switches are to be implemented in software by loading Pantou/Open WRT on a commercial wireless router or Access Point, which turns it to OpenFlow-enabled switch (OpenFlow, 2011); or loading Indigo on a physical switch with hardware features of Ethernet Switch ASIC's to run OpenFlow (Open Flow Hub, 2012). Library Automation System runs as a web-based software that implement library services as resources available for users on the computer network. Examples include KOHA, XLIB, etc.

Table 1: Security Policies and Configurations

S/N	Security Policies	Configurations
1.	Allow the BYOD Clients to connect to the network and use the automated library resources easily without the ability to acquire network information or perform managerial operation.	Configure an Access Control to Block the following protocol connections from the BYOD clients: • Telnet/Remote Connection • SNMP
2.	Allow only authorised network administrators to perform file transfer operation on the network.	Configure an Access Control to Block the following protocol connections from the BYOD clients: • TFTP • FTP
3.	Prevent Infected BYOD clients from generating excessive traffic that can hamper the speed of the network	Implement Bandwidth Management Application that allocates bandwidth to each user account of the BYOD clients and ensures that no user account exceeds it allocated bandwidth at any specific time.

4.1 Scope

Though the network model presented in this research work provides an open interface that can be exploited for innovative network functionalities, the security considerations in this work only focus on security threats posed by BYOD, as may affect the operation of the network that conveys automated library resources.

5. RESEARCH IMPLICATION

Library Automation Systems are web based systems that uses rights and privileges assigned to user account levels to control access to automated library resources it provides on the network. It neither caters for access to the network itself nor network security. Often, Library Automation Systems were deployed on a network with low security framework as emphasis is always placed on accessing automated library resources with less concerns to network security. This is because traditionally, Library Automation Systems are deployed on conventional network architecture involving devices (computers) owned by the learning institutions. Physical measures such as prevention of the users to use storage devices such as flash drives on computers, were sometimes put in place to prevent viral attack on the network. Therefore, implementing BYOD on the existing infrastructural design may generate security threats that can overwhelm the Library Automation System. On the other hand, the architecture presented in this research project, through the North-Bound Programmers Interface, supports the development of innovative network applications such as Intrusion Detection/ Intrusion Prevention Systems using advance methods such as Fuzzy Logic and Artificial Neural Network which can reactively mitigate security threats.

6. FINDINGS

This study reveals that the previous BYOD implementations reviewed does not address hardware incompatibility nor paucity of fund. ADA and Samsung Knox requires the development of a special network application to coordinate the Vendor-Based BYOD communication. Miradore Online requires licensing from the developers, which may be expensive to maintain.

7. EXPECTED RESULTS

The SDN Model presented in this paper, when implemented, will drastically reduce the financial burden on the Library Institution, limiting her responsibility to provision of network infrastructural devices such as servers, switches routers etc., while users come with their own device to connect to the network and access library automated resources. More so, since the network components in the architecture are OpenFlow-enabled, the problem of hardware incompatibility is eliminated. However, standardising policies formulation for the adoption of BYOD in Schools/Leaning Institutions remains a challenge recommended for future research.

REFERENCES

1. Andreas, B., Arsany, B., Martin, R., & Wolfgang, K. (2015) Survey on Network Virtualization Hypervisors for Software Defined Networking. arXiv:1506.07275v3 [cs.NI] 9 Oct 2015
2. Bruno, N., Marc, M, Xuan, N., Katia, O., & Thierry, T. (2014) A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. Communications Surveys and Tutorials, IEEE Communications Society, Institute of Electrical and Electronics Engineers. Retrieved Sep 2016 from https://hal.inria.fr/hal-00825087v5
3. Debasis, D. & Parnab, C. (2015) Library Automation: An Overview. International Journal of Research in Library Science, Volume 1 Issue 1
4. Device Administration (n.d.) Retrieved September 27, 2016 from https://developer.android.com/guide/topics/admin/device-admin.html
5. Glen, G. (2013) Reconfigurable Hardware for Software-Defined Networks. Department of Electrical Engineering, Stanford University. Retrieved September 27, 2016 from http://purl.stanford.edu/ns046rz4288
6. Miradore Ltd. (2016) Miradore Online. Retrieved September 27, 2016 from http://www.miradore.com/miradore-online/
7. OpenFlow (2011) Pantou: Openflow 1.0 for openwrt. Retrieved September 27, 2016 from http://www.openflow.org/wk/index.php/Open-Flow 1.0 for OpenWRT.
8. Open Flow Hub (2012) Indigo: Open source OpenFlow switches. Retrieved September 27, 2016 from http://www.openflowhub.org/display/Indigo/
9. Open Networking Foundation (2016) SDN Defined. Retrieved September 27, 2016 from http://opennetworking.org/
10. PDST Technology in Education (n.d.) Bring your own Device (BYOD) for Learning. Retrieved September 27, 2016 from www.pdsttechnologyineducation.ie/ictadvice
11. Rajput, P. & Gautam, J. (2010) Automation and problems in their implementation: An investigation of special libraries in Indore, India. International Journal of Library and Information Science Vol. 2(7), pp. 143-147
12. Samsung (2016) Knox Solutions. Retrieved September 27, 2016 from http://www.samsung.com/africa_en/business/solutions-services/knox-solutions/knox-workspace/knox-workspace
13. Shahaji, S. & Shri, G. (n.d.) Library Automation. Retrieved September 27, 2016 from http://eprints.rclis.org/22787/1/EPRINT%20Library%20Automation.pdf
14. Uddin, H. (2009) Library Automation: A study of the AIC, INSDOC and National Libraries of Bangladesh.
15. Zheng, C., Alan, L. & Eugene, T. (2010) Maestro: A System for Scalable Openflow Control. Technical Report TR10-08, Rice University.