Flexible network management and application service adaptability in Software Defined Wireless Sensor Networks

Kgotlaetsile Mathews Modieginyane[1], Reza Malekian[2], Babedi Betty Letswamotse[3]

[1, 2, 3] *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa,*

***Correspondence:** reza.malekian@up.ac.za Tel.: +27-12-420-4305*

Abstract The need for highly responsive and adaptable computing systems is essential in today's network computing age. This is principally due to the drastic evolution in broad computing platforms operating at highly descriptive and abstracted mediums such as; reconfigurable computing systems, smart automation systems, cognitive and parallel programming systems which communicate using very complex resources or modes. Hence, such systems must incorporate the best forms of technologies to cater for the rapidly growing and heterogeneously connected platforms such as with Internet of Things (IoT). However, to effectively manage these network platforms with such high-end computing resources, requires a well-structured and carefully implemented systems. This work implements a Software Defined Wireless Sensor Network (SDWSN) approach coupled with Discrete Event Simulation (DES) and a highly extensible and scalable Software Defined Networking (SDN) controller –OpenDayLight (ODL), to implement a software-oriented network environment to increase network service adaptability and simplify network management. The implemented approach uses the ODL's Model-Driven Service Abstraction Layer (MD-SAL) to facilitate the forwarding layer by applying state procedures to manage flow rules and introduce software-oriented network services. Experimental results indicate that in this approach, the traffic flow routing is significantly improved, with reduced transmission delays and that the underlying sensor nodes uses less energy since energy demanding tasks are performed on the controller.

Keywords: Internet of Things, Software Defined Wireless Sensor Networks, Discrete Event Simulation, Software Defined Networking, Model-Driven Service Abstraction Layer.

I. INTRODUCTION

Computing systems require the best level of technical capabilities to ensure accuracy and efficiency constantly as these aspects directly reflects the overall performance of a system. However, to achieve the best technological platform entails a lot of factors that must be considered when building any system. These factors include; the cost of hardware and software components to be used, the procedural concept as well as operations of the entire system, the task of developing and implementing this technology, the mission of testing the actual system for real life situations and enhancing it based on what the outcomes are. This analogy relates to every technological process that aims to develop any high-performance system or a computing platform. Such as in network computing systems, data centres and cloud computing environments. Highly responsive, scalable and adaptable systems are the order of today's technological development.

ISPs, data centres, industrial and manufacturing processing systems, mission critical systems, process control systems etc. constantly require responsive, automated, flexible and reliable system capabilities since efficient performance of these platforms are of extreme importance to consumers. However, to ensure customer satisfaction, entail a well-structured system which enforces and perpetuate data integrity, system connectivity and responsiveness. Data integrity and effective system response are the most essential factors in every mission critical system, especially which entails the purpose of security, monitoring and prediction, detection and alerting, etc.

Most importantly, the task of managing such systems require some high-level of technicality and skill to ensure system efficiency at all costs. This however, is not a simple task, especially in systems such as using Wireless Sensor Networks (WSNs). Even though such systems are extensively used in mission critical systems due to their simplicity of implementation and deployment, they have limitations that prohibits them to be actual choice of network computing for most purposes. To manage network traffic flow between the data control plane and the data forwarding plane, we use an OpenFlow communication protocol (OpenFlow version 1.3), since it is compatible with modern SDN controllers.

The OpenFlow (McKeown et al. 2008) communication protocol enables network personnel to effectively operate on network-wide traffic from the central SDN controller due to the decoupled nature of the data control and the data forwarding layers of the SDN architecture. It also promotes network flexibility and control (van Adrichem et al. 2014) as well as an opportunity for network computing innovation. This protocol standard makes it easy for network administrators not to be discouraged by proprietary network components whose purpose is only to forward data traffic. Hence, the OpenFlow protocol powers the southbound interface channel in terms of relaying data communication flows between underlying network devices and the SDN controller.

This work presents software-oriented approach to wireless computing sensor networks which allows rich application service adaptability and flexibility through network management. A framework to apply SDN strategies to WSNs, coiled as SDWSN; is implemented to enhance communications in WSNs (Li et al. 2018) by separating the control function from sensor network devices and as a result, improve energy efficiency [2] (Junli et al. 2017). In addition, WSNs offers the closest opportunity to achieve the connected world of every device [1] using diverse capabilities. As part of our approach, a DES tool is coupled with an SDN controller to ensure responsiveness, agility and network resilience as this will promote efficient network management and allow opportunities for network system innovation.

In this work, NS-3 (Network Simulator 3) and an OpenDayLight (ODL) controller are used as core tools of our experiments. Communication between the SDN controller and the modelled network is done through the OpenFlow protocol. To effectively achieve network flexibility and application adaptability, this work takes leverage in applying software-oriented strategies to the SDN controller and exploit the level of abstractions that can be achieved through NS-3 to introduce or project system strategic events on the network. This level of abstraction is used to also affect the actual output of the system as well as its response time. In this work C++ was used as a choice of modelling language due to its object-oriented nature and level of functional and application abstraction that it provides. Based on the applied approach, the following aspects are contributions of this work is:

i. Traffic flow routing has been significantly improved, with reduced transmission delays. As a result, the underlying sensor nodes used less energy since energy demanding tasks such as; network status checks, network abstracted services, integrating new network services, logging stored processes and communication management are performed on the controller.

ii. This approach has significant improvements the overall network performance for the implemented SDWSN system. Particularly in terms of network flexibility, resource management and network adaptability.

iii. Notable levels of flexibility have also been achieved, as SDN services such as remote access, network device management and simple updates could be done with ease.

iv. ODL provided simpler methods to support the SDN southbound interface to easily communicate with underlying network devices using its Service Chaining Functionality (SCF).

The rest of the paper is organised as follows; *Section II* outlines with depth and analysis, previous studies and achievement related to this work, *Section III* provides the strategic approach of the implemented model, tools and system parameters used, *Section IV* provides the actual methods followed and experiments performed in this work, with an analysis of all recorded results, *Section V* reflects on the achievements from this work as well as drawing attention to future research directions based on what has been experienced.

II. BACKGROUND AND RELATED WORK

Current technological systems are faced with high demands of powerful computation and networking capabilities. This is primarily because of the growth in technological sphere and the advancement in human or consumer demands for efficient and reliable systems. The research fraternity and industrial design and implementation are making considerable effort to meet these demands as well as expectations. As a result, this has increased the need to implement high performance and reliable computing systems in general. This has also influenced how researchers and developers look at future technology designs particularly in computing and networking. Next generation networks and computer systems promises to change the whole tradition of current technologies and their use. Much evidence has already been experienced with current focuses in SDN, IoT and cloud computing services as well as many other technological evolutions or ventures.

DES tools allows simulation of network components that operates at high level of system abstraction. It is however important to understand that in DES, events can transpire only on distinctive periods of the simulation thereby allowing other events to be processed or continue without being easily changed. Since the occurrence of an event or incident in a simulated environment causes a change in the system, it therefore critical to maintain and understand as to where in a simulation environment, the event must occur and for what purpose since any change affects the ultimate generated output. This work uses NS-3 as its simulations environment for WSNs. NS-3 allows discrete event simulation of several network models and routing protocols that can be modelled either using Python or C++ as its simulation modelling language.

OpenDayLight is an open source SDN oriented project that provides capabilities to manage and visualize network components (OpenDayLight 2017). To effectively manage our simulated network from a central point of view, an ODL controller was used. This is an SDN controller that offers a list of functionalities and capabilities since its development focuses primarily on SDN environments and network optimization. This controller was coupled with the simulation tool to render a visual representation of the system. ODL provides some level of programmability onto the network to change how computing networks function, process data and manage underlying network hardware and software components.

To compare and validate the efficiency of the OpenFlow protocol, the implemented system uses Border Gateway Protocol (BGP) and Network Configuration (NETCONF) protocols. BGP ensures that data is transmitted with minimum latency over the Internet between autonomous systems and provides alternative paths in case of route failure (Can and Demirbas 2016). NETCONF provides control and network management functionalities in SDN networks (Dallaglio et al. 2017). These protocols entail operational SDN strategies such as those of OpenFlow. Hence, they were applied in this work to compare their performance against OpenFlow.

The SDN model stands out in today's systematic developments due to its architectural implementation, data and process handling capabilities as this is envisioned to positively prepare a platform for network innovation (Modieginyane et al. 2017). This paradigm shift, implements a separate data control and traffic flow management to the underlying network components [5]. This makes it possible for SDN strategies to be application compatible to many technologies such as for wired and wireless systems. This work implements SDN strategies to WSNs with a focus to improve sensor network management in terms of access and configuration as well as the network application adaptability.

The main focus of SDN is to introduce application and functional programmability onto network computing systems. Hence, the application of SDN to WSNs has formulated the SDWSN approach. The principal idea of SDWSN is to enable WSN application technologies to be more efficient in terms of processing and network management. This approach, aims at simplifying wireless network and application policies in a way that these technologies can allow network innovation. An overview architecture of a SDWSN for a monitored environment is given if Figure 1 below.

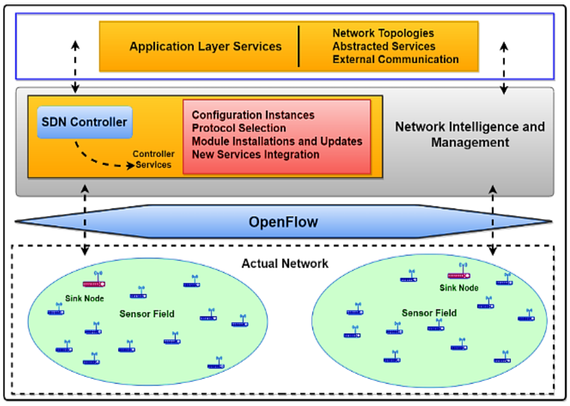


Fig. 1 An overview architecture of a SDWSN system.

Sensor aggregated data is relayed through OpenFlow capable switches to the central SDN controller, where all compute intensive tasks are operated. System administrator can accordingly operate on sensed data to support critical system requirements. This setup, also promotes network application creativity through abstracted network services. Figure 1 above, shows a sensor field with sensor nodes deployed for environmental sensing. Sensed data is relayed through OpenFlow channels to the controller for authoritative purposes. System administrator can remotely operate the network and install network functionalities that are relevant for every job that requires attention.

Numerous SDN approaches has been implemented to various network computing platforms to date including Sensor OpenFlow (SOF) (Luo et al. 2012), with emphasis to improve the adaptability of WSN applications for environmental monitoring. Their proposed model was reported to have significantly enhanced traffic load balancing as well as having reduced the amount of communication overhead. This approach has also reported significant improvements in the energy consumption of the system. An SDWSN prototype was proposed to improve the adaptability and the energy efficiency for WSN monitoring systems (Huang et al. 2015). In this work, an energy-efficient cognitive SDWSN prototype based on reinforced learning (RL) was developed for monitoring systems, wherein complex data fusion was managed centrally on the control plane while low complexity computations were performed on the data plane. Their results suggested that the proposed prototype had the ability to enhance self-adaptability of WSN environment monitoring with QoS. Moreover, by effectively hindering the transmission of unnecessary loads; energy efficiency is significantly improved, the amount of cross-plane communications was minimized and hence, load balancing in SDWSN was improved.

An adaptive error control implementation framework for SDWSN was proposed by Kadel et al. (2017). Their framework exploited the features of SDN and Forward Error Correction (FEC) to improve communication, reliability and flexibility in SDWSN. The framework is said to support adaptability at the transmitter and the receiver and also permits the use of FECs in various sections of the network or link. The authors claimed that the framework is capable of providing both iterative and non-iterative codes depending on the demand. Compared to non-iterative codes, the use of iterative codes offers more flexibility and adaptability.

Considering that link failure/ fault is one of the common issues in WSNs, Mukherjee et al. (2017) proposed a lightweight flow management model for reconfiguring flow entries in the flow table when link failure is experienced in the SDWSN forwarding plane. The authors claimed that their proposed model aimed to reduce control messages circulating over the network.

Significant results were also achieved by Wang et al. (2016) where an SDN routing protocol was implemented using OPNET to a multi-hop wireless network. This approach was reported to have increased the network lifetime compared to both Optimized Link State Routing (OLSR) and Ad hoc On-Demand Distance Vector (AODV) routing protocols, since it had the potential to provide the shortest path routing and disjoint multipath routing to wireless nodes.

The integration of SDN strategies to modern computing platforms serves as a potential direction to evolve them in that it allows capabilities for efficient control and understanding of all system processes and functionalities. Recent developments in the field of network computing are reporting tremendous achievements due to innovation capacities that are enabled by SDN. The effort of managing any system of networked components is a very difficult task.

However, the application of SDN to such systems has advanced some level of hope towards the implementations of high-performance technologies that will be easy to manage. Hence, with the right combination of tools and expertise, a lot can be achieved through SDN since it allows a platform for creativity. An SDWSN Control plane based on Constrained Application Protocol (CoAP) was proposed by Miguel et al. (2017). The authors provided extensive specifications of their proposed architecture with regards to communication infrastructure, control plane protocol, topology discovery and maintenance and flow control. The proposed control plane was deployed in Contiki OS and simulations were performed using Cooja. Their simulations measured control traffic overhead and the authors claimed that their proposed control plane showed potential due to the insignificant overhead introduced. However, the authors did not perform extensive tests to validate their claims.

Networking concepts and strategies such Network Function Virtualization (NFV), IoT and 5G have the same elements as that of the SDN architecture. Hence, these concepts can greatly benefit from the computing mechanisms proposed by the SDN project. It is possible to achieve some level of network resource orchestration, application automation and system function virtualization by correctly coupling and applying SDN strategies to modern and future network technologies. Previous network technologies relied primarily on circuit switching methods, whereas modern and future network technologies use both circuit and packet switching methods.

The use of OpenFlow communication standard and SDN strategies can greatly benefit from this tech technological perspective since traffic routing and network resource management are key in general networking systems. Effective network configuration and flexible network resource management can be realized since specific flow rules can be easily manipulated from the SDN controller, which directly communicate with OpenFlow capable devices.

SD-WISE- a fully operational Software Defined solution for Wireless Sensor and Actuator Networks (WSANS) and WSNs was presented by Anadiotis et al. (2017). The authors extensively described the major operations of SD-WISE and tested its effectiveness by demonstrating its features in various scenarios. The system is said to have extended SDN approach to WSNs and introduced some novelties as compared to its existing counterparts. Firstly, it introduced a more flexible way to define flows and the possibility to control the duty cycles of the sensor node radios in order to improve energy efficiency. Secondly, it supported NFV and geographic routing was implemented using NFV. Finally, SD-WISE exploited the tight interplay between trusted hardware and software to ensure that sensor nodes comply with the rules imposed by the remote recognised authority.

High demand for critical network services and rapid network responses are major concerns for system engineers and administrators, due to day-to-day customer needs. For a system to be customer satisfactory, some level of Quality of Service (QoS) must be guaranteed. However, to achieve this; requires a combination of high levels of system configuration and acute computational strategies. In the work by Nguyen et al. (2016) OpenFlow and SDN strategies were commended as potential solutions to deal with challenges experienced in wireless and switched networks towards the effort of realizing standard IoT platforms.

An SDN based sleep scheduling algorithm SDN-ECCKN that reduces the total time of transmission of a network was proposed by Xiang et al. (2016). Control nodes were selected to dynamically allocate different tasks. The control node selection was formulated as an NP-hard problem, taking the residual energy of the nodes and the transmission distance into consideration. As a result, an efficient particle swarm optimization algorithm was proposed to solve the NP-hard problem. The proposed optimization algorithm created a cluster structure in order to minimize the transmission distance and to enhance the energy consumption of the network. Their results suggested that the proposed algorithm outperformed its counterparts with respect to the network lifetime, number of functional nodes and number of isolated nodes and therefore, has potential to prolong the network lifetime. However, SDN-ECCKN experiences a high control overhead due to the continuous beacon messages that every node is required to send to the controller every time.

An SDWSN focused Situation Aware Protocol Switching Scheme (SAPS) for real-time support of application-specific requirements was proposed by Misra et al. (2017). Their proposed scheme consists of two phases: The decision phase- where supervised learning algorithms are employed to determine the appropriate protocol to be deployed; and the protocol deployment phase- where protocols are deployed at the sensor nodes, according to application-specific requirements. Their simulation results suggested that SAPS outperformed some of the already existing schemes with regards to delay, energy consumption and throughput in different scenarios. However, their scheme did not yield the best results with regards to Packet Delivery Ratio (PDR) and since the controller did not receive the network state in real-time it questions the real-time support that the authors aimed to achieve.

Other achievement such as in MonArch (Spachos et al. 2015), which is a Software Defined Infrastructure (SDI) framework, were recorded as having significantly tackled challenges experienced in cloud infrastructures. Their developed system achieved great results for carbon dioxide monitoring and alert using a wireless ad-hoc sensor network [3, 4].

III. SYSTEM MODEL

This implementation, uses NS3 for modelling the sensor network with parameter settings and options as illustrated in TABLE 1 below. Based on the proposed approach, OpenFlow (version 1.3) was used as the key communication protocol between the ODL controller and the modelled network. However, to evaluate its networking performance, the same setup used both NETCONF and BGP protocols. Virtual machine for both NS3 and the ODL controller were running on the Linux environment, with Ubuntu 16.04 Operating System (OS).

Table 1 Environment and system parameters

Parameter	Value by Type or Version
Node Mac/Phy Protocol	IEEE 802.15.4
Communication Protocol	OpenFlow 1.3
Network Size	400*900
Node Count	100 Sensor Nodes
Packet Size	128 bytes
Transmission Rate	250 kb/s
Simulation Time	60 (s)

A pictorial of the system setup is give in FIGURE 2 below. Both the ODL controller and the NS3 modelled network run on separate Virtual Machine (VM) instances. This allows the actual separation of the control plane and the simulated environment. In the whole system setup, the ODL controller is accessed as an external controller through the OpenFlow protocol. Simple network service such as; tracing the network, viewing and changing certain simulation parameters can be done from the controller. The controller also allows the view of the actual network nodes with connection statuses of all the network nodes.

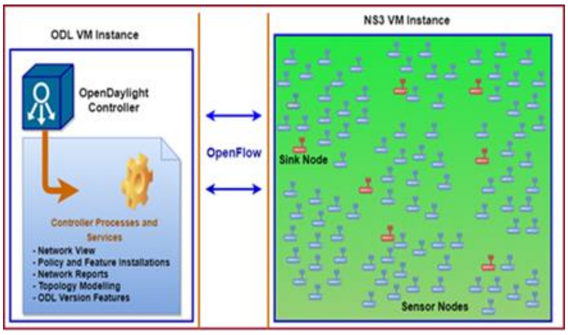


Fig. 2 Presentation of the network model.

The sink nodes are also illustrated on the NS3 VM instance in blue colour. These sink nodes, communicate with the simulated sensor clusters and cooperatively transmit aggregated traffic to the external controller via the OpenFlow channel. The ODL SDN controller is responsible for different network critical services such as; network reports, building network topology models, policy and feature installations.

Furthermore, the controller can be connected to application layer services through the northbound communication channel to associate or integrate application layer facilities or processes with the underlying network. Examples of application services can be; web-based applications, authorised remoted access to the system via the internet and other simple top layer Applications Programming Interface (API) applications. In our approach, flow state rules such as QoS roles performed on communication protocols, simple cluster-based functions executed on sink nodes and controller services such as routing and load balancing commands were implemented. To further explain our implementation.

To simplify and improve network management, the implemented approach uses the ODL's MD-SAL and the YANG models to apply system specific tasks such as; installing new network device features, discovery procedures, data and response handling services. To achieve these SDN controller services and functionalities this implantation takes leverage of the ODL's OFConfig (OpenFlow Configuration) rules to abstract OpenFlow network resources to influence their management as well as the YANG model's NETCONF services to manipulate and enable communication between controller services and network devices.

Throughput refers to the average communication channel's packet transmission rate with success delivery from the source to the destination device as expressed on the sensor nodes' packet radio measure. In this instance, the average flow transmission rate is measured on the three applied SDN communication protocols (*OpenFlow*, *BGP* and *NETCONF*) for different datagrams. The throughput measure is expressed from the communication protocol's capacity to successfully transmit nodes' sent packets over time.

IV. EXPERIMENTAL RESULTS

This section gives details with analysis of achieved results based on the experiments conducted with system parameters as tabulated in **TABLE 1** above. The main focus in terms of results analysis is directed to how SDN strategies improve WSN system applications in particular for monitoring applications.

To validate network flexibility and simplified resource management, the framework was tested by applying data manipulation and configuration operations on network resource. Service committing functions were the checked against the original status of network resources such as abstracted services and commanding network devices. Device access and response times for various network transactions were confirmed by checking changes in network behaviour.

Our experiments, indicates that using YANG models and MD-SAL services on the implanted approach, improved network resource access and manipulation. Services such as changing flow routing information and abstracted services –in our case checking cluster level status were performed with considerable simplicity and flexibility.

FIGURE 3 shows data aggregation plot for three different SDN controllers, whereas FIGURE 4 shows packet error rate over sensor traffic for different throughputs.

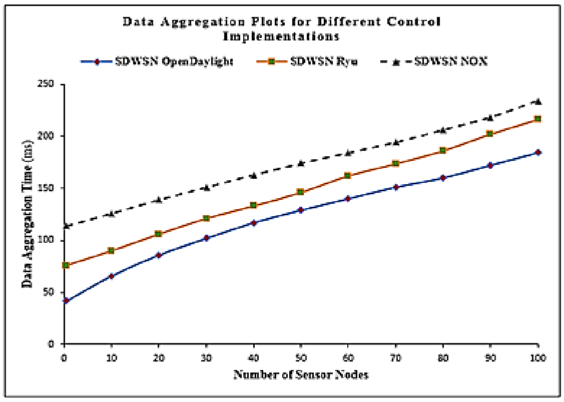


Fig. 3 Data aggregation plot for three different SDN controllers.

From FIGURE 3 above, it can be seen that; the systematic setup using ODL SDN controller performed best compared to both the implementations using Ryu and NOX SDN controllers. The ODL implementation produced lower aggregation times for data aggregation compared to both Ryu and NOX. This is a significant phenomenon by the ODL setup as it proves to have the least communication delay compared to the other two setups.

To further test the proposed implementation, a packet error estimation was also observed as in FIGURE 4 below, on separate setups using an external controller implementation (when the controller and the simulated network are on different Virtual Machines (VMs)) and a local controller implementation (when the controller and the simulated network are under the same VM and same network setup).

FIGURE 4 below, shows packet error rate plot for applied SDN capable communication protocols. SDN capable communication protocols considered for this work are; OpenFlow, BGP and NETCONF (in particular Request for Comments (RFC) 6241), since this version has SDN communication capabilities and can support proprietary network devices that have NETCONF interfaces.

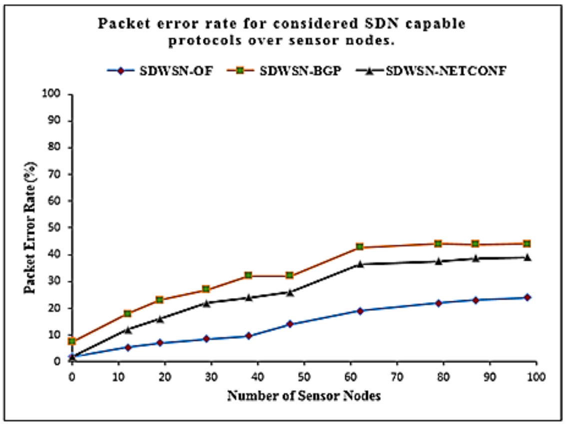


Fig. 4 Packet error rate per number of sensor nodes.

Our experimental results show BGP and NETCONF indicating almost the same performance for successful packet sent from the source device as compared to packet received at any destination device. In average, these two communication protocols indicate packet delivery rate of about 60%, whereas the OpenFlow protocol produced only 18% packet error rate. However, there is potential to optimize these protocols using SDN strategies according to system design requirements.

Based on our experiments, these protocols outperform each other on different network operations. For example, NETCONF performs better than both OpenFlow and BGP in terms of configuration speed. BGP is most efficient for traffic routing between base stations of gateway devices and the internet. OpenFlow promotes network innovation as proposed by the SDN strategy for abstract network computing. Hence, these communication technologies can be used together to achieve different goals within one network.

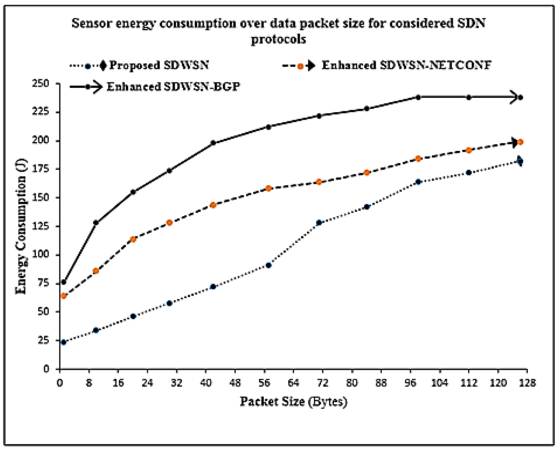


Fig. 5 Energy consumption for SDN protocols.

From above, FIGURE 5 shows a performance comparison for three different SDWSN implementation strategies for sensor nodes energy consumption versus data packet size. Performance comparisons were done amongst the proposed SDWSN, the enhanced SDWSN-NETCONF and the enhanced SDWSN-BGP strategies. The two enhanced versions of NETCCONF and BGP use the OpenFlow SDN data forwarding techniques as an enhancement approach. These two protocols uses QoS routing roles defined by the SDN controller as service parameters to ensure efficient transmission from the controller to data forwarding plane. Hence these protocols are SDN enhanced to support software-oriented strategies as defined by SDN framework. However, the proposed strategy indicated efficiency in energy utilization due to its simplicity of packet routing by the OpenFlow.

As opposed to this, the enhanced SDWSN-NETCONF showed better performance compared to the enhanced SDWSN-BGP approach in terms of energy consumption. This is mainly due to the processes that are involved in these protocols for packet or data transmission. On the other hand, the enhanced

BGP version showed remarkable results in terms of remote connectivity even though some challenges were experienced in terms of routing updates in the case were connectivity was lost. This also results in high energy utilization since targeted path updates should be rescanned to establish any defects.

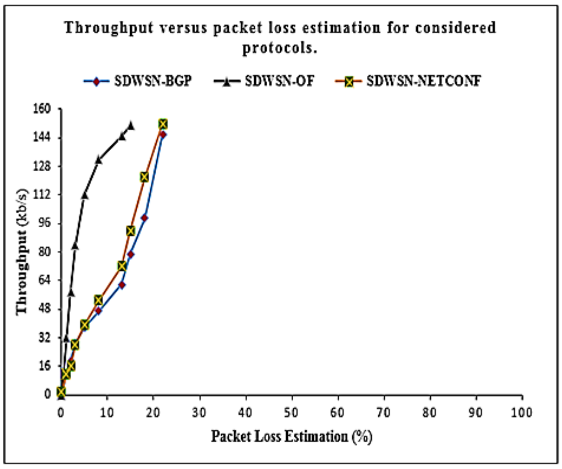


Fig. 6 Packet loss estimation over throughput.

FIGURE 6 shows a performance comparison difference in packet loss estimation over data throughput that becomes successively transmitted within the network from source to destination nodes. Yet again, the OpenFlow implementation of the proposed system produced the best results in terms of packet loss estimation compared to both the NETCONF and BGP system implementation.

The proposed system produced just about 14% packet loss estimation on average, whereas the NETCONF and BGP implementation produced almost the same rate of packet loss. On average the BGP implementation produced about 22% performed just below whereas the NETCONF implementation produced 20% packet loss when estimated.

This is a significant achievement by the OpenFlow implementation towards the overall improvement of WSN application systems. However, the NETCONF strategy showed a lot of potential for SDN applications especially in terms of device configuration.

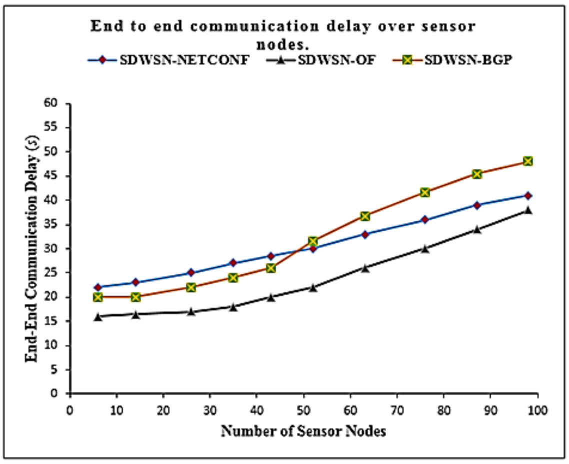


Fig. 7: End to end delay over sensor nodes.

FIGURE 7 shows a plot of associated end-to-end delay of the applied SDN communication protocols. The BGP implementation shows least performance however, almost the exact performance like that of the NETCONF implementation.

The NETCONF strategy produced an average of about 26 seconds end-to-end delay compared to the BGP implementation which produced end-to-end delay performance of about 34 seconds on average. The NETCONF implementation was observed to be almost constant in terms of delay for aggregated sensor data, whereas the BGP strategy seemed to be gradually experiencing some small amount of delay.

The OpenFlow implementation produced best results compared to both the NETCONF and BGP implementations with about 18 seconds. Another observation was that OpenFlow produced minimal increase in delay for high data aggregation of a larger number of sensor nodes.

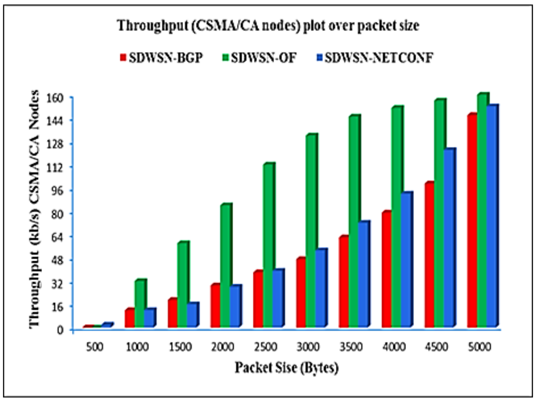


Fig. 8 Throughput plot over transmitted data.

FIGURE 8 shows a plot of throughput versus sensor transmitted packets. For sending packets over the communication channel, an experiment was performed with sensor nodes set to use Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) in an effort to avoid packet drops due to collision. Again, the OpenFlow implementation indicated high throughput support compared to both the NETCONF and BGP versions. This achievement by the OpenFlow shows significant performance towards its use for SDN based WSNs applications. However, improvements can still be done on this communication standard to enhance it for full scale SDN developments of any structure or design. On the other hand, NETCONF indicated incredible potential SDN communication tasks, especially for network configuration purposes. Even though in this instance BGP produced the least performance, it still holds significant capabilities for remote communication.

However, for very smaller packet sizes, transmission throughput tends to lower or decrease due to the transmission overhead introduced on the communication channel. This is because such smaller packet sizes, creates high transmission rate as they are quickly captured and transmitted through the channel and as a result creates serious communication overhead. In addition, systematic improvements on any computing scheme requires the best technologies for optimum performance, therefore, it remains the choice of the designer to opt for certain techniques over others for as long as best performances can be achieved.

Unlike previous works which mainly focused on energy management, communication overhead, load balancing and resource management, this work implements SDN communication mechanism with QoS role functionality to promote application abstraction for SDWSN systems. It further, applies software strategies as a driver to simplify network management and ensure network flexibility and adaptability.

V. CONCLUSIONS AND FUTURE WORK

This work proposed a SDWSN approach in an effort to improve the overall network performance especially for mission critical applications. NS3 network simulator was used for modelling a WSN, with an ODL controller used for the network intelligence. Two other SDN controllers – Ryu and NOX were also tested against the Implemented ODL controller to evaluate its performance. Performance comparison were done to effectively distinguish the best performing SDN controller and protocol for SDN developments in particular for this work. In terms of southbound communication performances, two SDN capable protocols namely; BGP and NETCONF, were used and compared with OpenFlow for our SDWSN implementation. Our experimental results produced best and least performances for different tests.

For controller performance efficiencies, ODL produced the best results, with Ryu coming second and NOX with the least performance. The ODL implementation produced lower aggregation times for data aggregation compared to both Ryu and NOX. Even though Ryu and NOX produced lower performances, they showed great potential in terms of managing network devices. Hence these SDN controllers can still be optimized to offer best performances. The ODL controller together with the OpenFlow protocol, produced the best results in terms of; data aggregation over sensor data transmission, packet delivery ratio, end-to-end delay time and energy utilization. The NETCONF protocol can also be a choice communication protocol for SDN application systems if it could be fully optimized for such, since this protocol produced promising results in all experiments conducted. This however, does not rule out Ryu as an alternate controller since it also has great potential for simple SDN applications. Otherwise it could be enhanced for high end SDN computing applications.

Our proposed approach showed significant improvements on the overall network performance for SDN based WSN applications. Particularly in terms of network flexibility resource management and network adaptability. Notable levels of flexibility have also been achieved, as SDN services such as remote access, network device management and simple updates could be done with ease. ODL provided simpler methods to support the SDN southbound interface to easily communicate with underlying network devices using its Service Chaining Functionality (SCF). However, this implementation can still be enhanced or optimized to fully support the SDN programmability as this will greatly promoted SDN adoption.

Since the SDN paradigm suggest that the controller be the global network intelligence with rich resources and control capabilities, it is mostly critical to have the communication protocols which supports simple but powerful interfacing abilities to both the application and infrastructure layers. Therefore, our future research directions will be aimed at software optimization opportunities for northbound interfaces for top layer applications. These applications can bring great improvements to SDN developments especially for high response systems and human interactive applications. Another direction is to implement cloud-based resources or applications that uses NFV and SDN strategies to host and run IoT infrastructures or applications.

ACKNOWLEDGEMENTS

We are grateful of the National Research Foundation (NRF) of South Africa as well as Telkom South Africa for their continuous financial support through this research work. We also acknowledge the University of Pretoria (UP) for lab resources that are provided to us for the success of this research project.

CONFLICT OF INTEREST

There are no conflicts of interest in this work. Every aspect of this work was a collective effort and agreement of all the authors herewith.

REFERENCES

Anadiotis AG, Galluccio L, Milardo S, Morabito G, Palazzo S (2017) SD-WISE: a Software-Defined WIreless SEnsor network. ArXiv: 1710.09147.

Can Z, Demirbas M (2016) Querying on federated sensor networks. Journal of Sensor and Actuator Networks. 2016, 5(3), 14, pp 1-15, doi: 10.3390/jsan5030014

Dallaglio M, Sambo N, Cugini F, Castoldi P (2017) Control and management of transponders with NETCONF and YANG. IEEE/OSA Journal of Optical Communications and Networking, 9(3): B43-B52, doi: 10.1364/JOCN.9.000B43

Huang R, Chu X, Zhang J, Hu YH (2015) Energy-efficient monitoring in Software Defined Wireless Sensor Networks using reinforcement learning: a prototype. International Journal of Distributed Sensor Networks, 2015(360428). http://dx.doi.org/10.1155/2015/360428

Junli F, Yawen W, Haibin S (2017) An improved energy-efficient routing algorithm in software define wireless sensor network. In: Signal Processing, Communications and Computing (ICSPCC), 2017 IEEE International

Conference on, pp 1-5. doi: 10.1109/ICSPCC.2017.8242610

Kadel R, Ahmed K, Nepal A (2017,) Adaptive error control code implementation framework for software defined wireless sensor network (SDWSN). In: Telecommunication Networks and Applications Conference (ITNAC), 2017 27th International, pp. 1-6. doi: 10.1109/ATNAC.2017.8215413

Li G, Guo S, Yang Y, Yang Y (2018) Traffic load minimization in software defined wireless sensor networks. Internet of Things Journal IEEE, PP (99):1-9. doi: 10.1109/JIOT.2018.2797906

Luo T, Tan H, Quek TQS (2012) Sensor OpenFlow: enabling software-defined wireless sensor networks. In: IEEE Communications Letters, 16(11):1896-1899, doi: 10.1109/LCOMM.2012.092812.121712

McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J (2008) OpenFlow: enabling innovation in campus networks. In: ACM SIGCOMM computer communication review, pp 69–74. doi:10.1145/1355734.1355746

Miguel MLF, Penna MC, Jamhour E, Pellenz E (2017) A CoAP based control plane for Software Defined Wireless Sensor Networks. Journal of Communications and Networks, 19 (6): 555-562. doi: 10.1109/JCN.2017.000095

Misra S, Bera S, Achuthananda MP, Pal SK, Obaidat MS (2017) Situation-aware protocol switching in Software-Defined Wireless Sensor Network systems. IEEE Systems Journal, PP (99):1-8. doi: 10.1109/JSYST.2017.2774284

Modieginyane KM, Letswamotse BB, Malekian R, Abu-Mahfouz AM (2017) "Software defined wireless sensor networks application opportunities for efficient network management: a survey," Computers and Electrical Engineering 000 (2017):1–14, http://dx.doi.org/10.1016/j.compeleceng.2017.02.026

Mukherjee M, Shu L, Zhao T, Wang D, Wang H (2017) Lightweight flow management for software-defined wireless sensor networks with link fault in data plane. In: Computer Communications Workshops (INFOCOM WKSHPS), 2017 IEEE Conference on, pp 998-999. doi: 10.1109/INFCOMW.2017.8116529

Nguyen TMC, Hoang DB, Chaczko Z (2016) Can SDN technology be transported to software-defined WSN/IoT? In: Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on, pp. 234-239. doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2016.63

OpenDayLight (2017), Platform overview. Available at: https://www.opendaylight.org/what-we-do/odl-platform-overview

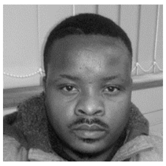
Spachos P, Lin J, Bannazadeh H, Leon-Garcia A (2015) Smart room monitoring through Wireless Sensor Networks in Software Defined Infrastructures, In: Cloud Networking (CloudNet), 2015 IEEE 4th International Conference, pp 216-218. Doi: 10.1109/CloudNet.2015.7335310

Van Adrichem NLM, Doerr C, Kuipers FA (2014) OpenNetMon: network monitoring in OpenFlow Software-Defined Networks. In: Network operations and management symposium (NOMS), 2014 IEEE, pp 5-9. doi: 10.1109/NOMS.2014.6838228

Wang J, Zhai P, Zhang Y, Shi L, Wu G, Shi X, Zhou P (2018) Software Defined Network routing in Wireless Sensor Network. In: Wan J. et al. (Eds) Cloud Computing, Security, Privacy in New Computing Environments. CloudComp 2016, SPNCE 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 197. doi: https://doi.org/10.1007/978-3-319-69605-8_1

Xiang W, Wang N and Y. Zhou, (2016) An energy-efficient routing algorithm for Software-Defined Wireless Sensor Networks. IEEE Sensors Journal, 16 (20): 7393-7400. doi: 10.1109/JSEN.2016.2585019

Biography



Kgotlaetsile Mathews Modieginyane; currently doing PhD in Electronics Engineering at University of Pretoria. Obtained MSc in Computer Science 2015, Hons Computer Science, 2012 and BSc in Computer Science and Electronics, 2010 all at the North West University. *Research Interests*: WSN, SDN, Signals Processing, Parallel Programing and Cloud Computing. He is also a member of SAICSIT.

Reza Malekian (M’10, SM’17) is an Associate Professor and Head of Advanced Sensor Networks Research Group at the Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, South Africa. His current research interests include advanced sensor networks, Internet of Things, and mobile communications. Prof. Malekian is also a Chartered Engineer and a Professional Member of the British Computer Society. He is an associate editor for the IEEE Internet of Things Journal and IEEE Transactions on Intelligent Transportation Systems as well as an editor for the IET Networks.

Babedi Betty Letswamotse received BSc Computer Science and Mathematics in 2011, Honors Computer Science in 2012 and MSc Computer Science in 2014, from North West University and is currently doing PhD in Electronic Engineering at University of Pretoria. *Research interests*: GPU general purpose computing, Automatic Speech Recognition, WSN and SDN. She is also a member of SAICSIT.