Computers and Electrical Engineering 000 (2017) 1–14

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Software defined wireless sensor networks application opportunities for efficient network management: A survey☆

Kgotlaetsile Mathews Modieginyane[a], Babedi Betty Letswamotse[a], Reza Malekian[a,*], Adnan M. Abu-Mahfouz[b]

[a] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa
[b] Council for Scientific and Industrial Research, Meraka Institute, South Africa

ARTICLE INFO

Article history:
Received 1 September 2016
Revised 27 February 2017
Accepted 27 February 2017
Available online xxx

Keywords:
Wireless sensor networks
Software defined networking
Internet of things
Software defined wireless sensor networks

ABSTRACT

Wireless Sensor Networks (WSNs) are commonly used information technologies of modern networking and computing platforms. Today's network computing applications are faced with a high demand of powerful network functionalities. Functional network reach is central to customer satisfaction such as in mobile networks and cloud computing environments. However, efficient management of WSNs remains a challenge, due to problems supplemental to them. Recent technology shift proposes Software Defined Networking (SDN) for improving computing networks. This review paper highlights application challenges faced by WSNs for monitored environments and those faced by the proposed approaches, as well as opportunities that can be realized on applications of WSNs using SDN. We also highlight Implementation considerations by focusing on critical aspects that should not be disregarded when attempting to improve network functionalities. We then propose a strategy for Software Defined Wireless Sensor Network (SDWSN) as an effort for application improvement in monitored environments.

1. Introduction

WSNs are sensor network technologies which are widely deployed on environmental monitoring, atmospheric monitoring, process monitoring, material sensing, security applications, etc. These networks operate on collective networking and computing of individual sensors based on their physical sensing properties and processing capabilities. Sensors nodes, cooperatively communicate and relay aggregated data to the main network control system for further processing and acting. In this regard, these sensors, must have an ability to conform to the collective networking functionalities as governed by their respective network policies.

In WSNs, sensor nodes can be randomly deployed, in essence allowing opportunities for applications even in inaccessible areas. This feature about sensor networks, allows the possibility of deploying a large number of sensors over intuited areas for as long as communications can be established and sustained among these sensor nodes. A WSN consists of, but not limited to; a WSN server, routers, switches, sensor nodes, etc. depending on the design setup as required for its purpose.

In this paper, we consider challenges experience in WSN applications. We provide brief introductions to both SDN and SDWSN and also highlight their technological prospects in WSN applications. This paper further considers challenges that are

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Zhihan Lu.

* Corresponding author.
E-mail addresses: reza.malekian@ieee.org, reza.malekian@up.ac.za (R. Malekian).

http://dx.doi.org/10.1016/j.compeleceng.2017.02.026

currently pointed out as being faced by both the SDN and SDWSN approaches, in question of whether they could improve general WSN applications. We also advice on network critical aspects such as security, reliability and scalability, since these are some of the most core features that needs to be carefully considered when planning to improve or optimize network functionalities.

This paper also reflects on issues surrounding the adoption of the SDN technology in general industry networks. Our inference towards these issues comes in the form of concerns and questions that needs to be looked at with careful considerations. In these concerns, we interrogate the SDN approach applied in WSNs as to whether this technology will introduce flexibility, improve the management as well as the overall performance of WSNs. We therefore point out that, through careful considerations of the advices provided by this survey, an enormous improvement in WSN technologies can be achieved.

The rest of the paper is divided as follows: Section 2 discusses WSNs use cases, Section 3 discusses WSN protocols and topologies; Section 4 discusses WSNs, their applications and limitations. Section 5 considers SDN and its improvement positions, the advancements of SDN into mobile and wireless networks, Section 6 discusses SDN standard protocol and security aspects. Section 7 introduces SDWSN, its related work, challenges and considerations regarding its adoption. Section 8 discusses future work. Finally, Section 9 provides concluding remarks of the survey.

2. WSN case studies

2.1. General perspective for designing WSN application systems

To design deployable and functional WSN systems, several factors and elements needs to be made in terms of case studies. As the same as with other systemic technologies, the development of each WSN application system is led by a requirement or need for that system. These requirements are at times influenced by technology markets or ongoing research approaches and developments. Following this phase, case studies are conducted to best understand the requirements, factors, elements, constraints, strategies and the feasibilities for implementing these technologies. For WSNs, these studies include but not limited to; understanding the size of the network to be designed, hardware and software requirements for the network design, the cost to implement the network as well as the resources to sustain and maintain that network. Other important case studies include; human related factors and the area wherein the network will be deployed.

2.2. Detailed factors for developing WSN application systems

Due to the technological market and research developments as described above, application systems need to solve a certain problem. This leads to a more detailed understanding and listing of technical requirements in terms of hardware and software needed to build these systems. These technical aspects include;1) the type of sensor hardware to be used, 2) operational characteristics of these sensors, 3) operational platform to deploy these application systems and 4) choice of routing protocols and considerations for operational standards, for both the network infrastructure, sensors, etc. Other technicalities to be considered include; mechanisms for power resources for these systems, data mining aspects, level of radio frequency operation and identification and scalability options for the network design.

3. WSNs routing protocols and network topologies

3.1. Routing protocols

To improve WSN system operations and applications, several access and routing protocols have been developed and applied which include but not limited to the following:

- *Medium Access Control* (MAC) protocol – whose main strategy is to reduce energy consumption, since sensor nodes in a WSN systems are battery powered thereby resulting in a limited network lifetime.
- *Low Energy Adaptive Clustering Hierarchy* (LEACH) protocol – which is the fundamental protocol to propose some level of data fusion as well as a focus to implement a strategy for low power utilisation in hierarchical WSNs.
- *Ad hoc On-Demand Vector* (AODV) routing protocol – its main objective is to reduce packets flooding which causes overhead within the network. One critical functionality of this protocol is to utilize routing tables to store routing information.
- *Sensor Protocols for Information via Negotiation* (SPIN) – These types of protocols are based on sensor nodes negotiations for allowing data transmission and resource adaptation mechanisms for energy saving.
- *Geographic and Energy Aware Routing* (GEAR) protocol – It is based on the energy and location of sensor nodes which are on their transmission paths towards their targeted regions. Its implementation facilitates the trade-off between energy and distance. Table 1 discusses the advantages and disadvantages of the protocols discussed above.

3.2. Network topologies

Commonly known WSN topologies are described below. Other forms of sensor network topologies are designed as a combination or extension of these network topologies due to the design requirements decided for different deployments.

Table 1
Advantages and disadvantages of WSN protocols.

Protocol	Advantages	Disadvantages
MAC		Energy wastage
LEACH	Increases Network Lifetime Energy saving due to aggregation by CHs	Dynamic clustering brings extra overhead Cannot ensure real load balancing in the case of sensor nodes with different amounts of initial energy Nodes use single-hop communication
AODV	Connection setup delay is less Routes are established on demand	Intermediate nodes can lead to inconsistent routes Multiple Route Request packets in response to a single Route Request packet can lead to heavy control overhead.
SPIN	Topological changes are localised	Not certain that the data will reach the destination or not. If the nodes that are interested in the data are far away from the source node and the nodes between source and destination are not interested in that data, such data will not be delivered to the destination at all. Idle nodes consume energy
GEAR	Increases Network Lifetime Helps in balancing energy consumption	

Table 2
Advantages and disadvantages of WSN topologies.

Topology	Advantages	Disadvantages
Star	Node failure does not affect other nodes Low power consumption	If central node fails, the whole network won't be able to work Low transmission range
Tree	Low power consumption High transmission range	Topology is very complex Not resilient to node failure Uneven power consumption across nodes
Mesh	High fault tolerance	High power consumption Increased latency Redundant paths
Hybrid	Low power consumption Fault tolerant Reliable communications	Expensive to implement Trouble shooting and maintenance is complex

I *Star topology*: – In this type of a network structure, sensor nodes transmit data only through a central device. Considered for small networks. A failure in any sensor node except the central device, does not affect the network or other sensor nodes. Not expensive to implement this architecture, as nodes share one central device.

II *Tree topology*: - Efficient for optimizing the power consumption or to extend the communication range of the network. Some level of scalability can be achieved in this type of a network. This architecture can be easily managed or maintained since the network is divided into branches or segments. If the root node fails, the lower structure also fails. This network design is not expensive to implement.

III *Mesh topology*: - All the sensor nodes within the same radio or communication range are connected. To reach a far destination device, data is transmitted through neighbouring nodes. Very fault tolerant since there are a lot of path options for other sensor nodes to successfully transmit data should one node fail. However, it is expensive to implement this type of a network.

IV *Heterogeneous topology*: - This results as a form of combination of two or more of the above topologies. This type of a network implementation is somehow, reliable, scalable, flexible and effective in its operation. However, it is expensive to implement, since it is a combination of two or more network topologies of different capabilities. Table 2 discusses the advantages and disadvantages of the topologies discussed above.

4. WSN challenges and deployment considerations

Even though WSNs are popular due to their simplicity of deployment and cost effectiveness, managing them is a difficult task pointing to their resource constrained nature. However, applications of WSNs have continued to grow regardless of the challenges experienced in their use. Basically, WSNs are envisioned to be deployed on large scale, as millions of wireless sensor nodes will be working cooperatively to transmit critical data and as well being connected to the internet as an effort for the realization of internet of things (IoT).

WSN technologies have been successfully developed and implemented in a wide range of applications as discussed before. These technologies are now commercialized according to their range and specificity of application. Table 3 illustrates some of the commonly used systems or technologies developed for critical applications.

Table 3
Common WSN applications and their technology systems.

Application	Condition	Parameters	Technologies
Agriculture	Planting: Soil preparations Animals: Moving patterns and behavior.	Soil: pH, Nutrients, Humidity, etc. Position, Animal Tracking, etc.	AgroSense, AgriServe, etc. ZebraNet, WiSense
Military	Surveillance [Safety and Security]	Infrared, Motion, etc.	Ultra-Stable Tripods, Radar, EcoKit, etc.
Health	Operative or Intensive Care	Heartbeat, Pulse, Rate, BP, Temperature, etc.	iMONNIT, Wisense, Autonomous Systems and Biomechatronics
Infrastructure	Surveillance, Control and Prediction, Measurements, Maintenance	Temperature, Motion, Infrared, Vibration, Strain, Stress, Air-Flow	Camera Systems, SensoNode, VIBCODE Transducer, etc.
Transportation, Automotive	Asset tracking, Presence	Radio Frequency, Accelerometer, etc.	Trackers, RFIDs, Smart Logistics, etc.
Water Quality	Observation and Control	pH, Iron, Nutrients, Color, etc.	Libelium Smart Water, etc.

Systems such as those used in health, military and environmental monitoring are some of the most commercialized technologies around the area WSNs. These technologies play a pivotal role in everyday living as some of them are mission critical systems. The wide adoption of WSN technologies is since these systems are affordable and not difficult to implement. However, the cost of implementing such technologies depends on their range (in terms of size and purpose) of operation. Due to factors, such as their affordability, ease to expand and not conforming to a particular network topology, WSN technologies could be adopted and optimized as the fundamental structure for the IoT network paradigm.

Commonly reported challenges include; sensor node energy limitations, low memory and processing capacity, low channel bandwidth and being application specific. A lot of work has been done particularly on energy limitations as an effort to improve the node energy utilization on WSNs. In addition to that, other methods of energy harvesting have been proposed as a means to leverage this limitation on sensor nodes. However, the energy issue is still a serious challenge in WSNs since it affects the lifetime of a WSN directly. Other challenges include high latency in communication, traffic congestion and processing delays on intermediate nodes due to the packet-based routing nature of WSNs.

Based on the proposed strategy, another controller functionality could be to write container rules or applications which will operate on certain network devices such as routers and programmable sink nodes. This functionality could be used to apply software focused strategies which will be responsible for load balancing, ensuring resourceful bandwidth utilization as well as perform quick search operations for data transmission to complement efficient processing, thereby reducing high delays.

Since WSNs are energy and processing constrained, important considerations must be made before the deployment phase of the network. These considerations includes, but not limited to; understanding phenomena/event requirements and the deployment environment (as this will assist in deciding on relevant equipment and their capability, and also whether there are no radio frequency disturbances within the area, sensor network connectivity, the ability of the network to self-configure in adverse or intrusion situation, choice of wireless protocol depending on the sensor type to be used as protocols/standards differ in power usability, throughput and communication range. We therefore put emphasis towards these considerations, so that if necessary; any software or resource oriented limitation be accounted for during the network planning phase.

5. SDN and its implementation challenges

5.1. Case studies for designing SDN

The concept of implementing SDN strategies to modern computer networks is imagined as a possible solution for common challenges experienced on them. However, due to the novelty of this networking paradigm, it remains unclear as to how this strategy can be fully applied to the effectiveness of today's networking technologies. In this paper, we carry out case studies for designing SDN strategies for most network environments. To start with, we discuss components that makes up a functional network, since these are critical when designing any network. A network is a platform of inter-linked, connected or interfaced hardware and software developed and operated by either human or machine capacity. It is therefore, an involving task to implement a functional network that uses new computing strategies such as SDN in production networks.

Case studies carried out in this paper, reveals some systematic and situational challenges that needs to be considered when designing SDN technologies. These includes; 1) The type and cost of infrastructure to host and support this new technology, 2) The capacity or skilled personnel required to operate or maintain the network, 3) Platforms and software needed to command the network, 4) Security levels and features which will be required for this technology. Other critical aspects that could be considered include; a) The risk of shifting to a new technology as opposed to optimizing the working old technology, b) The task of developing new procedures, protocols and applications which will support the new technology. It is therefore, important to understand and properly implement the functionalities of each component needed to support any new technology.

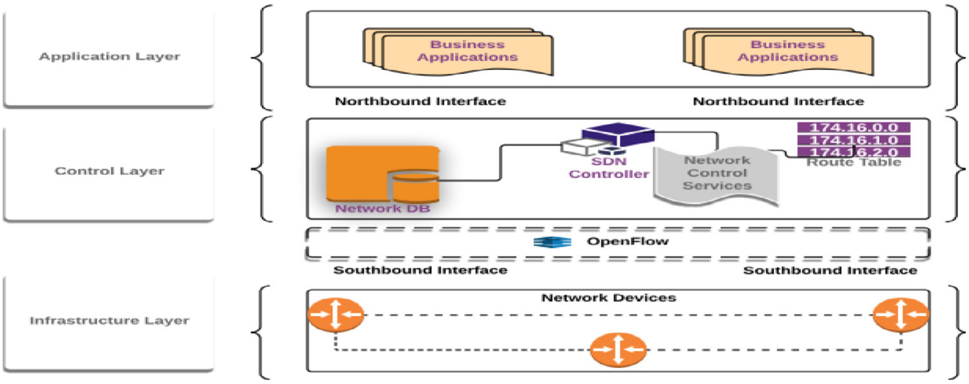

Fig. 1. An overview of SDN architecture.

5.2. SDN architecture

SDN is a framework that decouples the control plane and the data plane to allow network administrators to automatically and dynamically manage and control numerous network devices, services, topology, traffic paths, Quality of Service (QoS) and packet handling policies using high level programming languages and Application Programming Interfaces (APIs) of their choice [1].

We also describe an SDN approach using OpenFlow as shown in Fig. 1 above. OpenFlow by Open Networking Foundation (ONF); is a standard communications interface defined between the control and forwarding layers of an SDN architecture, which allows direct access to and manipulation of the forwarding plane of network devices such as routers and switches, both physical and virtual (hypervisor-based) [2].

Even though SDN applications promise to improve the overall networking experience by shifting complex processing functionalities from the network equipment to the controller, SDN itself has its own implementation challenges most especially on existing networks such as; infrastructural support for smooth transitions to SDN environments, flexible hardware for SDN deployment or applications, etc. However, as part of the proposed approach, this work intends to solve this problem by implementing controller functionalities that will be able to store current processing states and concurrently perform some automation on the base Operating System (OS) to support critical network devices for abstracted operations. This could also allow opportunities to implement virtual network components which will act as actual devices in performing other network state applications. Other concerns and challenges are that; a steady development is still needed to apply SDN concepts at network infrastructure level, since there is currently no general accord yet on how the programmable logic can be achieved, since there is currently no global standardization for SDN.

5.3. SDN challenges and opportunities in WSNs

Today's network computing applications are faced with a high demand of powerful network functionalities and performance. This high demand is mainly motivated by advancements in the area of computing spanning provider networks, data centers, cloud computing environments, etc. However, resourceful management of WSNs is a serious challenge, due to problems supplemental to them. Conversely, advancements and improvements in the manufacturing of computing or networking devices has awarded researchers and developers opportunities to experiment on these devices as an effort to enhance network computing capabilities and experience. Upon these concerns, recent network technology shift proposes SDN as a means to improve computing networks.

An improvement position is that, since resource management and process control are so difficult in WSNs due to their sensor properties and structural complexity, SDN aims at bringing convenient control mechanisms to WSNs. Current studies indicate that this capability will allow the controlling unit (the controller) to easily manage network resources as well as preparing a platform for other networking applications. One of the main objectives of SDN strategy is to ease up the process of network management. Primarily, SDN transforms a network administration problem into a simplified network programmable one. The suggestion is that by applying SDN, the complexity of sensor networks management can be simplified. With the flexibility gained by the introduction of SDN to sensor networks new protocols can be employed very easily without the need to reprogram or reconfigure any sensor node.

Distinctively, one of the core strategies of SDN in computing networks is to make an effort of visualizing the network architecture more clearly and comprehensive, as this could; 1) Make it easier for network operators to study and understand the network, 2) Allow operators to quickly come up with measures that are most relevant for specific network problems, 3) Encourage innovation in computing networks to prepare for or meet future demands, 4) Allow easy access of network resources so as to manipulate them as necessary and 5) Improve the way computing networks are observed or visualized and thus permitting network operators to easily manage and maintain adjacent network resources.

Please cite this article as: K.M. Modieginyane et al., Software defined wireless sensor networks application opportunities for efficient network management: A survey, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.02.026

Table 4
Comparison between an SDN network and a traditional network.

Potential	SDN	Traditional (Non-SDN) Network
Operate Features	Separation of the control and data forwarding planes. Easily controlled and deployed.	Hard-structured and logically coupled operations. Very complex control.
Implementation	Fast and easily implemented. Adapt to the need of application environment.	Time involving to implement. Operate on dedicated environments.
Stability	Currently unstable with few technological supports.	Stable with great network support.
Architecture	Software oriented with resource customizable features. Centralized network intelligence.	Hard-structured and operated on dedicated and proprietary devices.
Configuration	Configuration can be largely done remotely. Software monitoring and operation done centrally.	Network devices need to be configured directly and individually.
Management	Easily managed using APIs. Can be easily modified depending on the network demand.	Difficult to manage as network devices are solely proprietary and hard to access.
Maintenance	Can be easily maintained as new services or network upgrades can be done easily without affecting the whole network.	Difficult to maintain as the whole network might be affected by a small change in the network.
Error Checking	Easy to check network errors as software module are dedicated to do this. Therefore, errors can be quickly isolated easily from the network.	Error checking is extremely time consuming and difficult to isolate as operators have to the check whole network even for small errors.
Novelty	Network innovation becomes easy as new features of the network could be attached to the existing infrastructure. Does not need to revisit the whole network processing.	A simple change will require a serious study and understanding of the whole network structural design. Therefore, innovation is possible but difficult to implement.

A well compatible and structured SDN oriented network is aimed at achieving the mentioned strategic goals as an effort to improve the overall network computation. In this notion, an SDN based WSN is mostly anticipated to improve on its computing capabilities if of cause some SDN approach requirements are observed and met. Therefore, based on the settings proposed by SDN, the resource limitation nature of WSNs can be radically reduced.

5.4. SDN versus traditional network (Non-SDN network)

Traditional strategies of networking have proved to be capable and reliable for over the years. This points to the fact that large, medium and small size network architectures have been successfully implemented using these strategies to date. In today's technological demand for powerful systems, traditional ways of network computing face a tough time in terms of efficiency and management. In this regard network innovation becomes difficult to realize in traditional networks. This is mainly because, in addition to the network devices being proprietary, the network control is implemented on each forwarding device, thereby making it hard to access and operate on the functionalities of these hardware.

Some arguments still exist in terms of management and performance of SDN based networks and traditional network. These arguments are mainly looked at as comparisons between these two network technologies as an effort to highlight which technology brings benefits. These arguments are also further interpreted to the act of making decisions as to which technology could be used and for what intent that technology could be used. Table 4 gives these arguments in terms of comparisons.

The adoption of SDN in industrial networks remains a tough decision or even a hard step to take as SDN is still a new knowledge in computing networks and somewhat on the trial stages for adoption in such networks. A lot of work around SDN still needs to be done such as; issues regarding network device interoperability from different vendors, a comprehensive strategy for SDN infrastructure and at least an advocacy for secure SDN networks. We therefore infer that upon the realization of the above-mentioned aspects of SDN technology, the adoption of SDN as an effort for network computing improvement and innovation will be significantly advanced.

5.5. Wireless and mobile SDN

The concept of SDN was initially proposed for wired networks, as time went by and the concept proved to be successful on legacy wired networks, it also opened opportunities for other networks such as wireless, mobile and sensor networks to name a few. The introduction of SDN to these networks present a possibility of improving network coverage and connectivity and also reduce the cost of implementing changes when upgrading networks. Table 5 presents the already existing SDN based solution on mobile and wireless networks.

The introduction of SDN to wireless and mobile networks gave birth to some of the pioneering projects, one of the first efforts of combining SDN and Wide Local Area Networks (WLANs) was from the Cambridge University, and they deployed their SDN based solution called OpenRoads [3] on their campus network. OpenRoads is an innovation focused OpenFlow-based open source platform that separates network service from the underlying physical infrastructure, allowing free movement of users between any wireless infrastructures. SoftCOM [4] cloud-based, systemic future network architecture that influences the power of SDN and improves telecom operators' competitiveness by transforming the manner in which network

Table 5
Mobile and wireless SDN.

Security solution	Target environment	Network position		
		Core infrastructure	*Edge access*	*Carrier*
OpenRoads	Campus Wi-Fi network	Yes	–	–
OpenRadio	Campus/Cellular/Enterprise network	–	Yes	–
CellSDN	Cellular network	Yes	Yes	–
Odin	WLAN enterprise network	Yes	–	–
SoftRAN	Cellular network	–	Yes	–
SoftCell	Cellular network	Yes	–	–
OpenAPI	Enterprise network	–	Yes	–
SoftCOM	Heterogeneous network	Yes	Yes	–
MobileFlow	Cellular network	–	–	Yes
OpenRF	WLAN network	–	Yes	–
OpenRAN	Heterogeneous network	–	Yes	–

and business operations are managed. Odin [5] is a SDN wireless framework that introduces programmability to enterprise WLANs, it was envisaged to benefit enterprises by providing support in the form of authentication, load balancing, mobility, interference management, authorization and accountability.

In an effort to simplify data network management some projects were established, CellSDN [6] being the pioneer project. CellSDN provides scalable, fine-grain real time control with extensions. CellSDN comes with the benefit of introducing new saves easily at reduced cost. The network is easy to manage and easily incorporated with other wireless technology. The challenges of CellSDN are radio resource management, asymmetric edges, fine-grained policies and unplanned mobility. SoftCell [7] is a scalable architecture that supports fine-grained policies for mobile devices in cellular core networks, using commodity switches and servers. Through workload analysis and fine-grained policy support, SoftCell improves the flexibility and scalability of Long Term Evolution (LTE) networks.

Later on, OpenRadio [8] and SoftRAN [9] were proposed with the aim of introducing SDN innovation power to the wireless access domain. OpenRadio is a programmable wireless data plane that supports network evolvability, it provides modular and declarative programming interfaces across the entire wireless stack. OpenRadio allows various wireless protocols to be implemented on multicore platforms. It provides modular and declarative interfaces for programming wireless protocols and also provide support for application specific services. OpenRadio also comes with several challenges. Firstly, heavy decision planes would incur excessive performance bottleneck and inefficient utilization of wireless resources. Also, if a heavy sequential algorithm prohibits pipelined parallelism, logic-heavy blocks will no execute well on multi-core architecture. "Moreover, OpenRadio assumes decision/processing separation is significant in signal processing blocks. However, some blocks that need to take decisions, like packet-search block, are best expressed with some embedded decision code. If the gap in average and worst-case run times becomes too high, maintaining the deterministic semantic on that block could become very inefficient" [8].

SoftRAN is a software defined control plane for radio access networks that extracts all base stations as a virtual big base station. SoftRAN achieves simplified network management through plug and play control algorithms. It facilitates efficient utilization of wireless resources since the controller has the global view on the network load and the cell interference and also provides better mobility support and reduces costs. Other solutions to Radio Access Networks and cloud environment were also introduced. OpenRF [10] is a self-configuring SDN based cross layer architecture for managing Multiple Input Multiple Output (MIMO) signal processing, it permits commodity WIFI Aps to manage interferences in access points which are on the same channel. OpenRAN [11] is a cloud computing based Software Define Radio Access Network architecture for heterogeneous wireless networks. It provides open, controllable, flexible and evolvable wireless networks.

MobileFlow [12] an SDN based mobile network architecture for future carrier networks, its benefits are flow-based forwarding model that promotes a rich environment for innovation at the core of the mobile network. MobileFlow also provides support for fine-grained policy enforcement and is able to interact with Evolved Packed Core (EPC) network elements and software definitions of various mobile networks.

6. Security in SDN

Each and every systematic platform, structure or environment must entail some level of security. In general security is an integral part of every computing systems. Most importantly, security in these computing systems or devices must be considered and be accounted for during the planning phase of such systems. In addition to securing these systems, each system must be incorporated with intelligent security measures that the system should undertake at times when the system is compromised. We then emphasized that, as improvements in computing networks are such highly anticipated; security must be well thought of. Security in SDNs is still a major concern. In this regard, the following questions arises:

- How will security be implemented in SDN infrastructure?
- If implemented in a central point only (e.g. central controller), what will happen in the event where the controller is intruded?

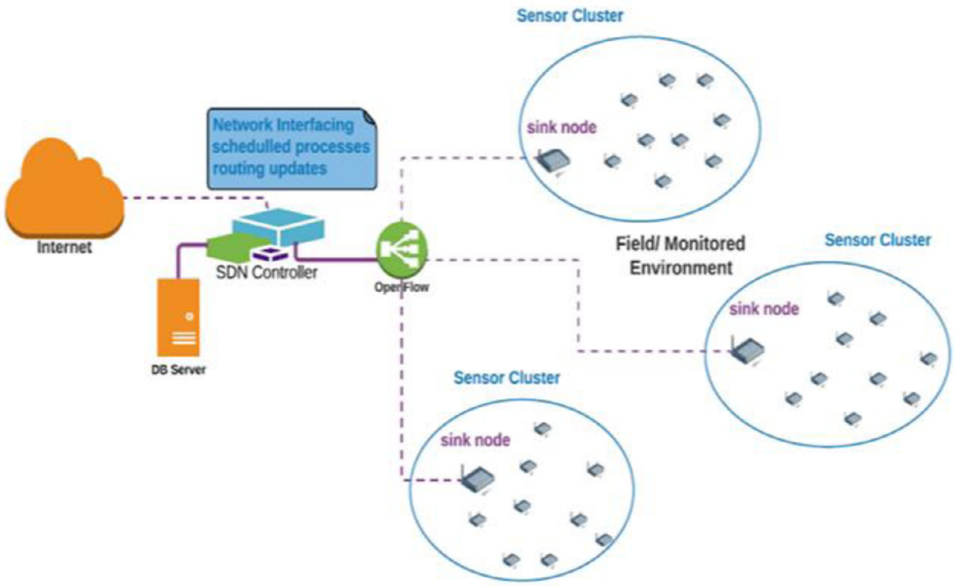

Fig. 2. Software defined wireless sensor network architecture.

- Should there be some level of security in every part of the SDN architecture?

These questions need to be answered through extensive research, development and system testing.

7. Software defined wireless sensor networks (SDWSN)

This section covers the SDWSN case studies, general overview of SDWSN, existing work, challenges and technological considerations regarding the adoption of the SDWSN concept.

7.1. Case studies for designing SDWSN application systems

Several technologies of WSN systems have been used for a wide range of applications due to their simplicity of implementation and cost effectiveness. Even though this has been the case, the high demand for extensively powerful systems of WSNs has rendered older technology systems and strategies to be less effective. This has therefore resulted in exploring other approaches for either optimizing current WSN application systems or developing new technologies. After conducting some literature survey around WSNs to improve their applications as well as testing some operations of WSNs applications through simulations, this work proposes a SDWSN system strategy as illustrated in Fig. 2 as a possible solution for challenges experienced in WSN application systems.

Case studies together with some simulation results have revealed some limitations in these types of systems even though they are operational to some extent. Technical limitations learned from the conducted studies include; a) Lack of innovation in current WSN strategies to cater for compute intensive applications that require highly responsive systems, b) Inability to support concurrent processes for different application needs, c) Lack of global access to operate different network devices, d) Highly involved tasks of detecting and troubleshooting faulty network points and e) Difficulty in deploying new efficient protocols in an effort to optimize the overall network performance.

7.2. Strategies for developing SDWSN systems: solving WSN challenges

The main drive to implement SDWSN is motivated by the SDN paradigm strategy to implement flexible and simple to manage computer networks. In addition to this, SDWSN strategy aims to implement networking technologies that allows the central view of the whole network. This paradigm change also introduces some level of programmability in production or application networks as a means to empower system administrators as well as network engineers to easily manage and optimize these networks.

As a direction to implement SDWSN systems that can solve WSN application systems challenges listed in 7.1, several key elements have been identified as possible solutions to decline these fundamental limitations such as: a) The need to develop software-oriented protocols for WSN systems which will be mandated to cater for different application processes on the fly, b) The process of developing software injection techniques to support programming abstractions for sensor networks - As this will influence the manufacturing platform for open programmable sensor, c) The course of develop highly encrypted software oriented algorithms to secure the control and data plane, d) To develop software strategies which will be responsible for data manipulation and presentation and e) To develop controller application strategies for accessing and updating sensor properties to best suite different application needs and operation parameters.

7.3. Overview of SDWSN

Due to the recent growth for sensor network applications in wireless communication and computing technologies, a lot of such application systems for WSNs have been prototyped and some partially developed and continue to be tested for efficient performance. However, some challenges such as function virtualizations and application automation remains to be serious concern to realize dynamic and fully efficient WSN applications. SDWSN is a potential solution to these problems since it proposes a programmable control that acts independent of the underlying network devices.

SDWSN refers to a networking paradigm that entails separating the control mechanism of an operational wireless sensor network from the data forwarding plane of the underlying vertically integrated network system. By introducing SDN techniques to WSNs, sensor nodes will only be performing forwarding tasks, whereas compute intensive tasks will be performed by the controller without affecting the overall energy consumption of the entire network. The benefit would be that the controller will not be resource constrained since it runs on a machine with more resources compared to the sensor nodes. Moreover, the controller will have the global view of the network and therefore being able to make optimal routing decisions based on the network's state information.

As part of an anticipation to advance WSN application systems that are secure by employing SDN computing strategies, the proposed architecture intends to develop and implement security algorithms as part of the controller functionality. These algorithms could be implemented on each level of the network through some form of function virtualization. Since the course of developing powerful security is an ongoing process, vigorous forms of system testing could also be used to exploit the system of any security weaknesses. This process will then inform future implementations of powerful security features.

SDWSN is also aimed at introducing a simplified implementation of a software oriented strategies for broad WSN computing systems by means of moderating sensor and other network resources workload for as far as data traffic and computing is concerned. An improvement position is that, since resource management and process control are so difficult in WSNs due to their sensor properties and structural complexity, SDN aims at bringing convenient control mechanisms to WSN by simplifying the whole network infrastructure. A SDWSN strategy as depicted in Fig. 2 consist of, a field sensor cluster which is deployed on a remote environment for phenomenal sensing such that the cluster is directly connected to a wireless sink sensor node. This architecture is comprised of; centralized SDN controller, OpenFlow Switch, sink node, sensor nodes forming a sensor cluster. The SDN controller maintains the global view of the whole network through the OpenFlow switch and other network devices, such that; the sink node and all the adjacent sensor nodes.

In this architecture, all the sensor nodes as well as the OpenFlow switch acts as data forwarding devices. The centralized SDN controller together with all the sink nodes, forms the control plane of the architecture. From the centralized controller through the OpenFlow switch, the sink node allows communication to certain sensors through some wireless transmission capability. Therefore, sensor clusters can be reached through this communication link for necessary sensor operations such as; updates, status check, sensor identification and isolation, etc. These sensors nodes must be compatible for systematical software manipulation and updating, thus to allow the SDN orientation.

Since the SDWSN proposes a centralized but flexible and programmable controller, simple network processes could be performed at the cluster level by implementing some network automation functionality on the sink node. These actions would be applied or performed based on requisitions or prompts at such levels (i.e. at sensor cluster level). This in some instances will allow the controller to be in a ready state for high processing demand applications of the network that needs efficient processing, such as routing and enforcing some QoS. Therefore, this would offload the controller from high data processing demands that should be maintained always.

Furthermore, to offload the SDWSN controller from compute intensive tasks, some level of automation could be implemented on the SDN customized sink node, where simple and low-level rules would be executed on the sink node. This will also increase the level of programmable functionality within the overall architecture, form a basis for on-demand application development at the network level and moreover, it would increase SDWSN innovation opportunities. Considering these possibilities, a simple sink-oriented sensor status rule for removing an inactive node on the cluster network is described in Fig. 3 in terms of a container, where status elements form a class objects of this container:

Some state rules depending on the application need could also be written to fulfil such processes either on the sink node or even on the switch. Simple status rules such as; updating the sensor data, introducing new functionalities either at switch-level or cluster-level, etc.

7.4. Existing work on SDWSN

At least some amount of work if not much has been done on SDWSN, the first attempt to combine SDN and WSN to solve the inherent problems experienced in WSNs was presented in [13]. The authors proposed a SDWSN architecture that clearly separates the control plane and the data plane. It also features Sensor OpenFlow which is the main component of SDWSN

Please cite this article as: K.M. Modieginyane et al., Software defined wireless sensor networks application opportunities for efficient network management: A survey, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.02.026

Algorithm 1: A Possible Simple Status Container Rule Implementation on the Sink Node

```
1:  Container_R          // Rule container for sensor node status.
2:  stateHolder { sensorID, remaining_Power, sensor_State }
3:  Do: Check the status of the sensor node

4:  StateRule_1: Use (2) to check the status of the sensor node.
5:  If (sensor node is inactive)
6:  Discard sensor node from the network - ( sensorRem sensorID.delete() ),
7:  Status update – Update the sink node - ( updSate update.node (sink) ),
8:  Status update - Update the global controller - (updState update.node(gController) ).

9:  StateRule_1: Use (2) to check the status of the sensor node.
10: If (sensor node is active)
11: Continue to perform data transmission – ( trans.data dest.node(sensorID) ).
```

Fig. 3. A simple container state rule implemented on the sink node.

as a standard communication protocol between the two planes. Later, SDWSN was used in Smart Grid WSNs [14] where it was said to have minimized complexity and power optimization in sensor nodes. The work in [15] proposed the use of SDN as a means to smartly manage WSNs. They suggested that the controller be placed at the base station and their paper argues that smart management using SDN envisions solving some of the inherent problem that come with WSNs.

One of the limitations experienced in WSNs is high sensor energy consumption. As a strategy, the concept of SDWSN intends to reduce high energy consumption on sensor nodes through some means of shifting other processing tasks to the SDN controller. Some work [16] attempted to solve this energy consumption issue by proposing a general SDWSN framework where the controller is placed at the base station and the sensor nodes performs only packet switching. They used OpenFlow as the core communication protocol between the controller and the switching elements. The proposed architecture minimized the energy consumption as anticipated but only to a certain level.

In [17], a SDWSN prototype was proposed to improve the adaptability and energy efficiency for WSN monitoring systems. In their work, an energy-efficient cognitive SDWSN prototype based on reinforced learning (RL) was developed for monitoring systems, wherein complex data fusion is managed centrally on the control plane while low complexity computations are done on the data plane. Wang et al. [18] proposed a SDN based sleep scheduling algorithm SDN-ECCKN that reduces the total time of transmission of a network at the same time maintaining the network connectivity. In their work, computations are performed by the controller and there is no sensor-to-sensor propagation in their algorithm. Their results suggest that SDN-ECCKN shows significant improvement compared to Energy Consumed Uniformly-Connected K-Neighborhood (ECCKN) considering network lifetime, number of functional nodes and number of isolated nodes.

Due to the application-specific nature of WSNs, it remains a challenge to manipulate its sensor nodes after a full system deployment as these sensors are built for their specific applications. In this regard, a work by Miyazaki et al. [19] proposes an SDWSN approach to curb this challenge by means of reconfigurable sensor nodes composed of an ultra-low power Field Programmable Gate Array (FPGA) and a Microcontroller Unit (MCU) through role assignment. In their proposition, a wireless communication was used to inject roles on certain sensor nodes, thereby allowing the possibility to manipulate or even update the functionality of these sensor nodes depending on the system's application requirement. They reported that their system produced some extent of network flexibility and indicated some potential to serve as a standard structure for WSN deployments depending on different user application requirements.

Another SDWSN attempt to improve traffic routing and WSN sensor programmability is that from the work in [20] which introduced an SDN solution for WSN systems, intended for reducing the amount of information exchange between the SDN controller and the adjacent sensor nodes and for enabling these sensor nodes to be programed as finite state machines. They reported that their system approach increased network elasticity and provided simplified network programmability since it allowed system developers freedom to use programming languages of their choice when implementing the SDN controller. Joint Routing and Resource control (JRRC) protocol for Software Defined Sensor Networks (SDSNs) which rearranges routes and allocate resources for new applications and network services in real-time to maximize the overall throughput was proposed in [21]. Their simulations reveal that the controller is capable of allocating resources reasonably to maximize the network throughput.

Authors in [22] investigated the possibility of an SDN based localization algorithm, whereby they proposed a Crameo-Rao Lower Bound (CRLB) based localization node selection algorithm. They formulated an integer linear programming problem based on energy satisfaction by using the controller's network global intelligence. The most reliable nodes for localization were chosen by calculating the contribution of each anchor node in the CRLB metric. Their results indicated that their SDN based localization algorithm achieves considerable improvements. In our view this adds to be a great improvement in the aspect of WSN network programmability especially with the ability to use any high-level programming language when implementing the SDN controller, since the controller build language choice is one of the aspects highlighted in this paper. Even though SDWSN is regarded as a potential solution for some limitations experienced in WSNs, there is still a

Please cite this article as: K.M. Modieginyane et al., Software defined wireless sensor networks application opportunities for efficient network management: A survey, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.02.026

serious lack of available strategies or implementations of clear and stateful SDWSN developments, as much as there is little literature regarding this venture.

Some few developments in this field, have at least proposed on some strategies of OSs as well as on the possibility of implementing multiple controller architecture. However, clear technologies in this venture are yet to be done. Of course, ongoing research using different simulation platforms is currently the hope to achieving the best outcome of this proposed strategy to improve WSN technologies. A work in [23] has proposed a method of testing new protocols presented as TinySDN tool. Their proclamation is that, this tool enables some SDWSN functionality using TinyOS. They also presented that their architecture provides an opportunity of having multiple controllers for SDWSN. Authors in [24] proposed TinySDM - a Software Defined Measurement (SDM) architecture for WSNs. TinySDM has numerous abilities such as providing support for conducting various measurement tasks, allowing easy customization of various measurement task and allowing efficient deployment of new measurement tasks. They implemented TinySDM on the TinyOS platform and evaluated its performance in a testbed comprised of 60 nodes. Their results suggest that TinySDM is flexible, efficient and easily programmable.

In another attempt on applying the SDWSN concept as a solution to WSN challenges, the authors in [25] proposed an approach of TinySDN, a hardware independent TinyOS based SDWSN nodes framework that enables multiple controllers within the wireless sensor network. It incorporates the SDN enabled sensor nodes and a programmable SDN controller. This framework aimed at addressing; leveraging in-band control, higher communication latency, smaller link layer frames and limited energy supply. Their results revealed that TinySDN enables the achievement of flexibility provided by the SDN concept, while it also introduced a memory overhead but it does not hinder features related to WSN applications.

One of the main functionalities for SDWSN is to accredit simple-programmability as well as to simplify WSN management for any applicable commitment – that is; being able to implement some level of programming on these sensor networks and at the same time being able to easily manipulate them for any network demand. Moreover, SDWSN deployments or systems must be less or considerably expensive since this architecture promotes a system where less sophisticated network components such as more routers and switches are used. Some work in this area have considered sensor network adaptability measures depending on the prompts or the demands lifted by the network, whereas other proposals focused on optimizing nodes functionalities by implementing their systems using sensor nodes that can be configured with some level of programmability. Some technological achievements have been realized such as in the work by Miyazaki et al. [26] where software defined sensor nodes which can be configured for dynamic operation, changes due to sensor specific task requirements.

A new approach that takes advantage of the SDN-WISE (-WIreless SEnsor) envisioned state information to support QoS provision in WSNs was proposed by [27]. Their approach is based on using a state to report the level of congestion on each node to the controller. SensorSDN is a novel SDN based architecture for WSNs that can be used for various IoT systems which was proposed by [28]. Firstly, they proposed new control plane services for reinforcing automatic topology discovery, sensor mobility, sensor virtualization and managing network policies. Moreover, they proposed SDN based customizable flow tables on existing Low-Rate Wireless Personal Area Networks (LR-WPAN) technologies to meet the requirements of different sensor packets. Finally, their architecture allows a programmable cross-layer optimization between the MAC layer and the network layer as well as data aggregation to support fine grained flow processing.

Flauzac et al. [29] proposed and application of SDN in WSNs with a structured and hierarchical management. They argued that their approach of applying structured and hierarchical SDN, promises to solve some of the existing problems in WSN management. Their work proposed a cluster based architecture with multiple base stations as hosts for performing SDN control functions and also as cluster heads at the same time. They further proposed a software defined cluster sensor network (SDCSN), a general architecture where the controller can communicate with other SDN domains through some border controllers.

7.5. Challenges and considerations

One critical challenge in SDWSN is the efficient assignment of spectrum resources to the virtual network, which then contributes to the spectrum resource problem. However, dynamic programming and graph theory based spectrum sharing algorithm is proposed in [30], wherein a performance improvement in this regard was achieved. Once more, there is a concern as to say; since SDN propose to facilitate a control of the underlying network from a dedicated point, it is alleged that the whole network will collapse in case where the controller is disrupted or somehow fail. Also, with the current envision about Internet of Things through WSNs [31,32], it must be considered as to whether a relevant infrastructure can be realized on SDWSN.

Subsequent to uncertainties that surrounds the standard architecture as well as the effective adoption of SDN technologies in production or application networks, some level of understanding is needed regarding its structural components and the fundamental benefits that SDN proposes to bring. Due to the distinctive architectural model of SDN, network customization [33] and resource optimization in WSNs could be achieved using this approach, thereby improving the overall network performance. SDN is also aimed at providing network stability and flexibility in WSNs, through the enhancement of some of the critical network aspects such as; process scheduling, traffic routing, resource access, network abstraction and programmability. Congruently, SDWSN aims at improving resource utilization and “open” network programmability in WSNs.

8. Future directions: technology and research opportunities

Looking at the radical improvements and changes in today's network computing as well as reported achievements in this field, it is most certain that in the future more sophisticated but easy to use and manage technologies will be realized.

Some level of attention needs to be drawn towards the SDN/OpenFlow northbound and southbound interfaces as these parts of the framework are of empirical means to the overall technological approach. We project that these interfaces still need to be further explored for efficient network understanding as well as resource optimization. In terms of future, we have identified the following research opportunities; 1) SDN controller virtualization for multi-controller sensor clusters, 2) Enhanced global network experience: Improvements on northbound interface communication, 3) Southbound interface optimization for efficient device access and controller communication and 4) SDN strategies for runtime and computational overhead in sensor clusters.

9. Conclusions

This paper looked at challenges experienced in WSNs as well as making critical analysis as to why careful considerations must be made before the deployment of WSNs especially for monitoring applications. Critical factors affecting applications of WSNs have also been discussed with advices that needs to be considered towards the network planning phase. We have also proposed a method of implementing simple state rules on the sink node as an effort to improve the SDWSN programmability as well as to offload the controller of such low-level compute tasks. As a technological position, information regarding current developments using SDN techniques have been provided.

Based on the studies conducted in this survey, we envision the SDN approach in WSNs to be a promising direction, as this approach will extremely evolve these application systems.

Conflicts of interest

All the authors declare that there are no conflicts of interest in this work.

Acknowledgements

The authors would like to acknowledge the National Research Foundation (NRF) of South Africa (grant number: RDYR160404161474 and IFR160118156967) through their excellent support both in motivation and funding to the realization of this ongoing research venture.

References

[1] Jain R. Introduction to software defined networking (SDN); 2013. available at: http://www.cse.wustl.edu/~jain/cse570-13/ .
[2] McKeown N, Anderson T, Balakrishnan H, Parulkar GM, Peterson LL, Rexford J, et al. OpenFlow: enabling innovation in campus networks. Comput Commun Rev 2008;38(2):69–74.
[3] Yap KK, Kobayashi M, Sherwood R, Huang TY, Chan M, Handigol N, et al. OpenRoads: empowering research in mobile networks. SIGCOMM Comput Commun Rev 2010;40(1):125–6.
[4] Libin D, Huawei Technologies Co, Ltd,. SoftCOM: Reshaping the future of network architecture. Huawei Global Analyst Summit; 2013. online: http://www.huawei.com/ucmf/groups/public/documents/attachments/hw_260647.pdf .
[5] Suresh L, Schulz-Zander J, Merz R, Feldmann A, Vazao T. Towards programmable enterprise WLANs with Odin. In: Proc. the first workshop on hot topics in software defined networks services HotSDN '12; 2012. p. 115–20.
[6] Li LE, Mao Z, Rexford J. "Toward software-defined cellular networks. In: European workshop on software defined networking (EWSDN); 2012. p. 7–12. 2012.
[7] Jin X, Li LE, Vanbever L, Rexford J. Softcell: scalable and flexible cellular core network architecture. In: Proc. the Ninth ACM conference on emerging networking experiments and technologies services CoNEXT '13; 2013. p. 163–74.
[8] Bansal M, Mehlman J, Katti S, Levis P. OpenRadio: a programmable wireless data plane. In: Proc. the first workshop on hot topics in software defined networks services HotSDN '12; 2012. p. 109–14.
[9] Gudipati A, Perry D, Li LE, Katti S. SoftRAN: software defined radio access network. In: Proc. the second ACM SIGCOMM workshop on hot topics in software defined networking services HotSDN '13; 2013. p. 25–30.
[10] Kumar S, Cifuentes D, Gollakota S, Katabi D. Bringing cross-layer MIMO to today's Wireless Lans. SIGCOMM Comput Commun Rev 2013;43(4):387–98.
[11] Yang M, Li Y, Jin D, Su L, Ma S, Zeng L. OpenRAN: a software-defined RAN architecture via virtualization. SIGCOMM Comput Commun Rev 2013;43(4):549–50.
[12] Pentikousis K, Wang Y, Hu W. Mobileflow: toward software-defined mobile networks. IEEE Commun Mag 2013;51(7):44–53.
[13] Luo T, Tan H, Quek TQS. Sensor OpenFlow: enabling software-defined wireless sensor networks. IEEE Commun Lett 2012;16(November(11)):1896–9 26.
[14] Sayyed R, Kundu S, Warty C, Nema S. Resource optimization using software defined networking for smart grid wireless sensor network. In: 3rd International conference on eco-friendly computing and communication systems (ICECCS); 2014. p. 200–5. 2014.
[15] De Gante A, Aslan M, Matrawy A. Smart wireless sensor network management based on software-defined networking. 27th biennal symposium on communications (QBSC) Kingston, ON; 2014. pg. 71-75, 1-4.
[16] Jayashreel P, Infant Princy F. Leveraging SDN to conserve energy in WSN-An analysis. In: 3rd International conference on signal processing, communication and networking (ICSCN); 2015. p. 1–6. 26 - 28, 2015.
[17] Huang R, Chu X, Zhang J, Hen Hu Y. Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: a prototype. Int J Distrib Sens Netw 2015;2015:12 Article ID 360428. doi:10.1155/2015/360428.
[18] Wang Y, Chen H, Wu X, Shu L. An energy-efficient SDN based sleep scheduling algorithm for WSNs. Elsevier. J Netw Comput Appl 2016;59:39–45.
[19] Miyazaki T, Yamaguchi S, Kobayashi K, Kitamichi J, Guo S, Tsukahara T, et al. A software defined wireless sensor network. In: International conference on computing, networking and communications (ICNC); 2014. p. 847–52. 3-6, 2014.

[20] Galluccio L, Milardo S, Morabito G, Palazzo S. SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks. In: IEEE conference on computer communications (INFOCOM); 2015. p. 513–21. 26 - 1, 2015.
[21] Li M, Li J, Shen L. Joint routing and resource control in software-defined sensor networks. In: International conference on wireless communications & signal processing (WCSP); 2015. p. 1–5. 15-17, 2015.
[22] Zhu Y, Zhang Y, Xia W, Shen L. A software-defined network based node selection algorithm in WSN localization. In: IEEE 83rd vehicular technology conference (VTC Spring); 2016. p. 1–5. 15-18, 2016.
[23] B.T. de Oliveira, C.B. Margi, "TinySDN: Enabling tinyOS to software-defined wireless sensor networks", http://sbrc2016.ufba.br/downloads/Salao_Ferramentas/154318.pdf
[24] Cao C, Luo L, Gao Y, Dong W, Chen C. TinySDM: software defined measurement in wireless sensor networks. In: 15th ACM/IEEE international conference on information processing in sensor networks (IPSN); 2016. p. 1–12. 11-14, 2016.
[25] de Oliveira BT, Gabriel LB, Margi CB. TinySDN: enabling multiple controllers for software-defined wireless software networks. IEEE Lat Am Trans 2015;13(11):3690–6.
[26] Miyazaki T. Dynamic function alternation to realize robust wireless sensor network. Int J Handheld Comput Res 2012;3(3):17–34.
[27] Di Dio P, Faraci S, Galluccioz L, Milardoy S, Morabitoz G, Palazzoz S, et al. Exploiting state information to support QoS in software-defined WSNs. MedHocNet 2016 2016.
[28] Hakiri A, Gokhale A. Rethinking the design of LR-WPAN IoT systems with software-defined networking. In: International conference on distributed computing in sensor systems; 2016. p. 238–43. 26-28, 2016.
[29] Flauzac O, Gonzalez C, Nolot F. SDN based architecture for clustered WSN. In: 9th International conference on innovative mobile and internet services in ubiquitous computing; 2015. p. 342–7. 8-10, 2015.
[30] Yang M, Li Y, Jin D, Su L, Zeng L. Opportunistic spectrum sharing in software defined wireless network. IEEE J Syst Eng Electron 2014;25(6):934–41.
[31] Wang Z, Ye N, Wang R, Li P. TMicroscope: Behavior perception based on the slightest RFID tag motion. Elektronika ir Elektrotechnika 2016;22(2):114–22.
[32] Wannenburg J. Body sensor network for mobile health monitoring, a diagnosis and anticipating system. IEEE Sensors Journal 2015;15(12):6839–52.
[33] Cloete NA, Nair L. Design of smart sensors for real-time water quality monitoring. IEEE Access 2016;4:3975–90.

Kgotlaetsile Mathews Modieginyane currently doing PhD in Electronics 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: WSNs, SDN, Signals Processing, Parallel Programing and Cloud Computing.

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. Currently doing PhD in Electronics at University of Pretoria. Research interests: GPU general purpose computing, Automatic Speech Recognition, WSNs and SDN.

Reza Malekian is an Associate Professor at the Department of Electrical, Electronic, and Computer Engineering, University of Pretoria. His current research interests include advanced sensor networks, Internet of Things, and mobile communications. Reza is a Chartered Engineer and also an associate editor for the IEEE Internet of Things Journal and IEEE Access as well as editor of the IET Networks.

Adnan Abu-Mahfouz received MEng and PhD degrees in computer engineering from the University of Pretoria. His research interests are software-defined wireless sensor networks, network management, network security, low-power and long-range wireless sensor networks.