

Efficient topology discovery protocol using IT-SDN for software-defined wireless sensor network

Joseph Kipongo, Ebenezer Esenogho, Theo G. Swart

Department of Electrical and Electronic Engineering Science, Center for Telecommunications, University of Johannesburg, Auckland Park, South Africa

Article Info

Article history:

Received Oct 25, 2020

Revised Jun 14, 2021

Accepted Dec 16, 2021

Keywords:

Discovery protocol

Energy efficiency

Software-defined networking

Wireless sensor networks

Topology discovery

ABSTRACT

The internet of things (IoT) and wireless sensor networks (WSNs) are two promising technologies for supporting new services and applications. However, because IoT and WSN entail interaction and transaction of device topology discovery protocols, WSN facing some challenges. Topology discovery (TD) is important for WSN in IoT since sensor nodes (SNs) are the main points of this network. Networking these SN (IoT-devices) has a few difficulties such as energy and bandwidth consumption, and data storage issues resulting from frequent interactions and transactions between devices. Because these challenges cannot be resolved by one solution, we focused on reducing the energy consumption associated with a WSN. We proposed software-defined networking (SDN) model to tackle energy-efficiency issue in a WSN setup by integrating SDN and WSN, which gave rise to a more robust system, software-defined wireless sensor network (SDWSN). In this direction, we proposed an improved fuzzy-logic-based strategy, fuzzy topology discovery protocol (FTDP). It used IT-SDN (an SDN-based WSN framework) to increase the network's lifespan due to its low energy consumption. The link layer discovery protocol (LLDP) to build the SDN controller's topology. The system's performance was presented and compared, demonstrating that with an effective SDWSN discovery policy, energy efficiency is achievable.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Theo G. Swart

Department of Electrical and Electronic Engineering Science, Centre for Telecommunications

University of Johannesburg, P. O. Box 524, Auckland Park, 2006, South Africa

Email: ebenezere@uj.ac.za

1. INTRODUCTION

Background; the advent of the fourth industrial revolution (4IR) has brought to bear the relevance and optimal usage of WSNs. Since new protocols that enable devices, appliances, industrial processes, and machines to connect and communicate have been developed, its applicability on the internet of things (IoT) and industrial internet of things (IIoT) cannot be over-emphasized, especially in this era of artificial intelligence (AI). WSN is the foundation upon which nodes connect, be it devices, industrial processes, or machines. However, IoT refers to an interconnected system of smart devices for data collection and data transfer through wireless communication without the interference of humans. WSN has a significant function in IoT where the sensor nodes (SNs) are the main actors in the SDWSN architecture. Studies have shown that WSN offers several great benefits, such as wireless personal area networks (WPANs), flexibility, and scalability [1]. Nevertheless, it also has several drawbacks, such as bandwidth and energy consumption, data storage, and packet collision problems. To alleviate some of these challenges, the SDN is proposed to address issues such as energy and forwarding decisions. This is made possible by the SDN controller, which plays the

role of a manager/coordinator [2]. It does that by removing dissipating energy from a node to a controller that is centralized in SDN [2]-[5]. Such duties like management and routing are taken care of at the controller while abstraction and network function visualization (NFV) are dealt with at the application level [5].

A process where an entity gathers information of topology in the same network is called topology discovery (TD). Topology management (TM) is still important for network management, where the application plane is facilitated by the controller. The controller finds a topology through the host, switches, and through interconnected connections between openflow (OF) switches [6]. The controller from the switch discovers the host by getting an information-in message. The efficacy of the OF discovery protocol (OFDP) lies in its ability to discover the OF switches during the handshake process with the controller and interconnected links of the OF switches. The requirement for an efficient TD in SDWSN is predicated on the fact that, since WSN is dynamic, there is a chance that as a result of the network node mobility, more information is generated and transmitted to the controller for an updated topology [7]. These interactions and transactions create overhead traffic that affects the network's resources negatively. The network TD plays an important role in the system's energy level, which could be favorable or unfavorable to the network. Most often, overheard traffic among the ordinary nodes and the control plane leads to higher energy consumption. On the other hand, few nodes consume less energy when the communication among the nodes is long [5], [6]. For the TD to function properly, it must adhere to some of the requirements of the controller. Firstly, the TD should be error-free. This is because a topology error can bring a routing that is wrong to the neighboring node. When there is a routing error, this can be disastrous and lead to unnecessary energy consumption. Secondly, an efficient TD is needed in the sense that the controller should not be flooded by the TD with information that is not important. Hence, when there is topology events information, the TD must send that information to the controller.

Being the central coordinator allowed by the TM, the SDN controller is there to maintain the operations in the application plane, where network traffic is routed over the network topology to its target. The topology is discovered by the controller through host discovery and switch discovery. It identifies the host by getting a data-in message from the switch and then creates links between the switches [1], [8]. During that link process through the controller, there is switch discovery and interconnections among switches are found through the OFDP [7]-[9]. WSN applications are not too demanding in terms of information rate. Hence, an efficient SDN is needed when using the sensor resources, even if we have lower information rates with such efficiency [9], [10].

This study's main contribution is to upgrade the execution of SDWSN, by introducing a fuzzy logic-based controller using IT-SDN to increase the lifespan of the network and improving the efficiency of the SDWSN. TD uses fuzzy theory where every sensor node must give the necessary data containing the energy, the neighbors' list, and the node cost (NC) [11], [12]. Figure 1 shows the fuzzy topology discovery protocol (FTDP) design in SDWSN.

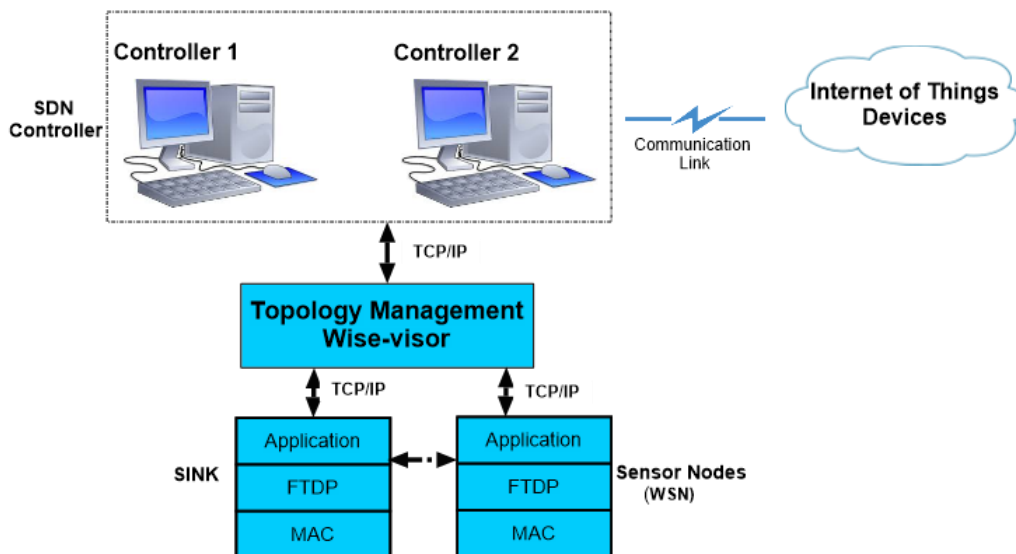


Figure 1. FTDP architecture in SDWSN

Thus, a fuzzy logic-based controller considers various network properties [13], such as remaining energy (RE), number of neighbors, and network traffic, to choose the best forwarding sensor node and

improving the network's execution. The link layer discovery protocol (LLDP) is a type of TD protocol under the name of IEEE 802.1AB previously referred to as station and media access control connectivity discovery (SMACCD) [14]. The network devices use LLDP to reveal their identities, functions, and their list of neighbors. LLDP is used to extract the topology information and to report it to the controller without modifying existing protocols, resulting in the conservation of energy.

Problem statement; due to the challenges of the traditional WSN such as energy consumption; there is a need to design a suitable TD protocol for SDWSN approach and reduces the overhead traffic among ordinary nodes and the control plane, which leads to higher energy consumption. This will be done by developing a new TD to determine the list of neighbors, the present traffic, and the amount of the RE. Based on the above-mentioned problem the following sub-problems are identified, The WSN topology keeps changing when the nodes are mobile and the processing power as well as the memory is dynamically assigned. Thus, there is a need to investigate to what extent these changes have on the TD. The majority of existing WSN topology control algorithms is known to be energy-inefficient and yet the SDN features have recently emerged to support scalable TDs. Thus, there is a need to investigate and develop an energy-efficient scalable TD algorithm for SDWSN.

Related work; SDN focuses on supporting network management and improving its scalability [15]. In these networks, there is a separation between the control and data plane. Software-defined sensor networks (SDSN) [16] are the ones focusing on the controller sensor server which has the responsibility for telling what the sensor node should do in the network. Related work that focuses on programming fields is published in [17]. The authors gave three main layers for application-programming interfaces (APIs), namely implementing information, observing network state, and maintenance of SN connection.

The authors proposed tiny software-defined measurement (TSDM) [18]. It is an SDN measurement architecture for WSNs using a language similar to usual C programming named tiny code language (TCL). For TSDM, task measurement and the transfer of another coding function can be handled easily by just moving the binary code. OF presented in [9], [15], is one of the implementations of SDN solutions where the SNs in OF deal with a new information in flow tables [19].

Jayashree and Princy [20] presented a WSN-based structure, where the OF protocol is used by the controller. The purpose of the research in [20], is to deal with the main problem of WSNs, which is energy efficiency. In this research, SDWSN is used to deal with traffic problems, like energy consumption, and the collection and transfer of information. Sensor OF (SOF) presented in [21], which is the same as OF, is the SDWSN's main component, which is an important protocol of communication between the control and data/information plane. With the help of SDWSN, the network becomes programmable using SOF when dealing with flow table on every SN. The TSDN presented in [22] helps promoting tiny operating system (TinyOS) to play the role of SDWSN's nodes. Each SDN-enabled SN identifies its neighboring nodes and sends information to the node of the SDN-controller over the collection tree protocol. One of its main functions is topology information collection. SDN-enabled SNs transmit numerous information by waiting for feedback. When getting that feedback, the information from those SN, having the neighboring sensor nodes ID, is included in the flow table. The authors have introduced a routing technique that is flexible for WSN to improve the network life span by utilizing a fuzzy logic framework [23]. The algorithm of that routing technique utilizes the RE and the best selection of the node for the optimization of energy routing in WSN [24], [25] introduced Soft-WSN, which is an architecture of SDWSN to deal with IoT's application requirements. The authors proposed an SN and a controller to enable SDWSN. They presented two controller parts like the topology manager (TM) and device manager. The TM supports specific devices duties, like tasks' and delay's sensing, and sleep/dynamic scheduling. In addition, the device manager supports network topology to address the network's quality of service (QoS).

SDN solution for wireless sensor network (SDN-WISE) presented in [26], is one of the new innovations in this field that supports packets or data to collect topology information for SDWSN [27]. It is one of the SDWSN's solutions that was used for the implementation of the approach called the fuzzy logic protocol. Fuzzy-logic protocol in SDN-WISE was proposed in [28], but the authors used a network simulator (NS2) which does not give clear simulation results in terms of the energy consumption and delivery rate as compared to IT-SDN platform. The importance of IT-SDN is to give a clear separation of the FTDP protocol to achieve southbound communication, to discover the neighboring nodes and the controllers. In [29], the authors designed a fuzzy controller for energy efficiency using multicasting in SDWSN.

In the SDWSN study, a fuzzy strategy is used for the TD, which is a significant aspect for TD and routing choice in SDN-based WSNs. The presented strategy gives a better WSNs TM by separating controller logic from the SN in SDN, and by utilizing the fuzzy logic framework when the topology is changing in WSNs. There are different methodologies about TD in SDWSN from various papers, with their qualities, shortcomings, and goals as shown in Table 1.

Table 1. Comparison of the methods

Methods	Goals	Protocols	Advantages	Drawbacks
Azzouni <i>et al.</i> [30].	Actualizing sOFTDP as a TD module in Floodlight	LLDP	Bidirectional forwarding detection. Hashed LLDP content.	Efficiency and security problems of OFDP
Ochoa-Aday <i>et al.</i> [31].	OF-based network and crossover network.	BDDP and LLDP	Dynamic configurations and innovations in the network. Diminution of the controller.	Difficulty to use in the TD. Complication to help with OF switches.
Loweckamp <i>et al.</i> [32].	Producing an algorithm of the physical topology of the Ethernet network.	“Spoofed” ICMP-echo packets	Data scaling. Sufficient information.	No forwarding database. Dumb switches. Acceptable traffic. Insufficient data.
Donnet <i>et al.</i> [11].	Decreasing redundancy in the network.	Doubletree algorithm	The tree-like structure of routes on the internet. Doubletree can decrease estimation load.	Absence of thought of productivity for web checking framework. The issues of redundancy bandwidth consumption Restricted nearby data.
Breitbart <i>et al.</i> [3].	Identify the new physical topology of an IP network	MAC addresses	Managing inadequate data and VLANs.	The straightforwardness of components across convention layers. Heterogeneity of network elements.
Donnet <i>et al.</i> [33].	Establishing the performance of doubletree, reduce transmission overhead and increase probing effectiveness	Doubletree algorithm	Decreased correspondence costs	Higher traffic levels. Hindrance to doubletree's usage. Inborn scaling issues.
Roy Choudhury <i>et al.</i> [34].	Topology data collection and distribution of the network.	Stigmergic transmission Link stability. Data deteriorating	The handling of the sensor network in TD	Node mobility Information convergence Navigation strategies
Bejerano <i>et al.</i> [35].	Partial topology information demonstration using SNMP MIB	Skeleton-path refinement algorithm	Strong completeness	Complexity for physical TD.
Pakzad <i>et al.</i> [36].	TD overhead traffic reduction. Config the largest possible network	OFDPv2 (OF discovery protocol version 2)	Higher control traffic overhead high data rates	Scalability and reliability issues.
Abdolmaleki <i>et al.</i> [28].	Growth of the delivery rate of data, decrease of the packet misfortune, and energy increase.	Fuzzy TD protocol (FTDP).	Higher the delivery rate of packets, lower data casualty, higher RE of the network.	Insufficient energy utilization load balancing, dynamic topology and node mobility

About the issues of scalability, as mentioned in [36], the researchers presented the approach called OFDPv2. This approach modifies the behavior of the controller and introduces another rule on every switch, that indicates that every LLDP information collected from the controller should be transferred to all available ports. This method has the advantage of decreasing overhead traffic from the controller when using the OFDPv2 approach. The scaling issue is additionally mentioned in [33] since the goal is determining *Doubletree's* performance when utilizing *Bloom filter* have the advantage of decreasing the cost of communication. The issues presented in [28], like energy consumption, were solved using the FTDP approach. The advantage of that approach is the improvement of the SDWSN performance and decreasing the information's delivery rate, and improving the energy of the network.

2. METHODOLOGIES/ALGORITHMS

2.1. SDN-WISE topology discovery protocol

In this section, an overview of the SDN-WISE solution is provided by giving the specifications in the SDN-WISE design, and a brief overview of the SDN-WISE technical approach is provided as well as the TD protocol.

2.1.1. Specifications

The principal reason for presenting SDN into WSNs is presented on the new paradigm called SDWSN as already discussed. Studies show that WSN has poor efficiency in terms of energy, memory, and data processing. Thus, SDN-WISE should be efficient when dealing with the sensor, irrespective of other factors. SDN-WISE supports the duty cycle when it comes to energy efficiency if the radio interface is turned off and should also support data/information aggregation [14]. These characteristics were not part of the

scenarios of OF. Moreover, the communication between the controller and SN should not be taken into consideration for the accomplishment of the efficiency of the system. SDN-WISE nodes can also handle packets in their header and payload composition. The SDN-WISE architecture is illustrated in Figure 2. In the data plane (the left-hand side), there is the forwarding layer (like the TD), in-network packet processing (INPP), and the forwarding component (FWD). For the IEEE 802.15.4 standards, there is the media access control (MAC) and the physical (PHY) layers. The control plane is the right-hand side where there are SNs that do not synchronize with WSN.

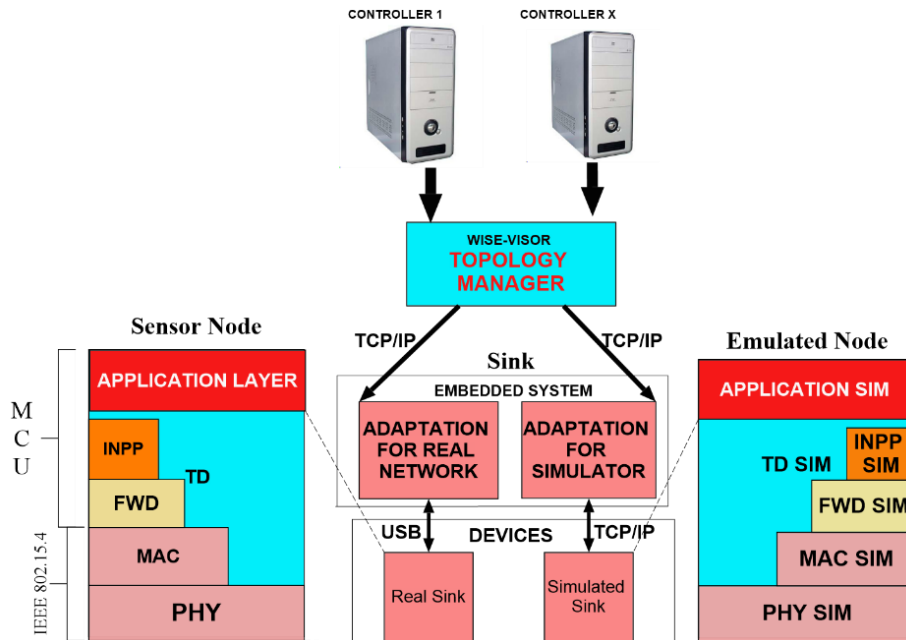


Figure 2. SDN-WISE protocol stack

The forwarding layer has the purpose to support the information in the control plane. The INPP has the responsibility to collect information for implementing the functions of in-network information. Moreover, the responsibility of the TD is to discover the neighboring nodes and send information to the controller. In the control plane, there is one controller node called the WISE-Visor that has a layer called TM, which outlines the resources of the network. Hence, the performance of various controllers on similar nodes gives different logical networks.

2.1.2. Approach

For a better protocol, the TD layer will run SDN-WISE protocol stack and its functions are data exchange and data processing of the TD packets. The information that those packets have is about the hops' number and battery level. When a node gets a TD packet, there is a selection and comparison of nodes to determine the one that has the best hops' number. These packets are utilized to organize a list of WISE neighbors since it has the addresses of each neighboring node and the battery level. The SDN-WISE table is a packet structure of the SN since every WISE flow-table entry has a section of matching rules that indicates the entry's conditions to be applied. For the new TD information, the SDN-WISE flow table must be cleared and reconstructed for an update of the topology.

2.1.3. Protocol framework

SN and at least a sink are the characteristics of SDN-WISE networks [26]. The Sinks are the entry for the SN managing the data plane and the control plane's elements. The SN runs the protocol stack of the data plane as illustrated in Figure 2.

The SN has two sections such as a micro-control unit (MCU) and an IEEE 802.15.4 transceiver [26]. Over the IEEE 802.15.4 protocol stack, the forwarding layer manages the MCU that supports the new packets as mentioned in the WISE flow table. The in-network packet processing (INPP) layer has the function of data aggregation. In the implementation of SDN-WISE, the INPP layer connects packets that are small, and which should be sent with similar paths. Doing that will decrease the network overhead. The TD

layer touches all layers of the protocol stack via application programming interfaces (APIs). Therefore, the controller helps data to be collected from the nodes of the TD layer. This layer gives an application layer, which enhances the IEEE 802 APIs [26].

In the control plane, there are one or more controllers, for example, the WISE-Visor. It has a TM layer that has an entry to APIs given by all the layers. TM layer has two specifications, which are data collection from the nodes and data transfer to the controllers. The other specification is to control all protocol stack layers as mentioned to the controllers which run in a similar node that has the TM layer. The Emulated Node is the last layer that can connect with a sink that is the non-real sink. In this layer, a TD SIM layer touches all other layers.

2.1.4. Topology discovery protocol

The TD protocol controls all SN and is important for developing data and sending it to the WISE-Visor. The information is kept in the TD protocol from the next hop (the devices' number of a portion of information that travels from its source to its destination) close to the controllers and controls its current neighboring node to be updated. For this reason, every sink in the SDN-WISE network sends a TD information over a wireless transmission channel periodically. Such information has the characteristic of the sink that made it. From SN *B* getting a TD packet for an SN *A*, there are the following activities via pseudo code/algorithm 1.

Algorithm 1.

Input: All nodes from source to sinks

Output: Determine each node location

Initialization:

1. Adds the battery level of SN *B*.
 2. Find the current neighboring nodes of SN *B* as well as its RSSI.
 3. **If** there is only the RSSI of SN *B* and the updated energy.
 4. **Then** node *B* has collected TD information with a smaller value of the range from the sink.
 5. **ElseIf** Those conditions are not satisfied.
 6. **Then** node *A* resets the value stated in the TD information to the current value.
 7. **Else** its next hop is set to the Controllers like *B*.
 8. **If** the level of energy is set by node *A* in the similar topology of the TD information.
 9. **Then** send the updated TD information over the wireless transmission channel.
 10. **Else** each SN delivers information that has the current neighboring nodes and sends it to the Wise-Visor.
 11. there is a cleared neighboring nodes' list.
 12. **End If**
 13. **End If**
-

2.2. Topology discovery for SDWSN using LLDP

LLDP is a neighbor discovery (ND) protocol of a single hop [31]. Its function is to show its capabilities and getting the information that is similar from the neighboring OF switches. The most useful component of LLDP in TD is OF-based networks. The control plane is required to build the TD for gathering important information about the forwarding plane and the devices' link. This process will automatically detect when another switch takes part in the topology. The discovery procedure is shown in Figure 3.

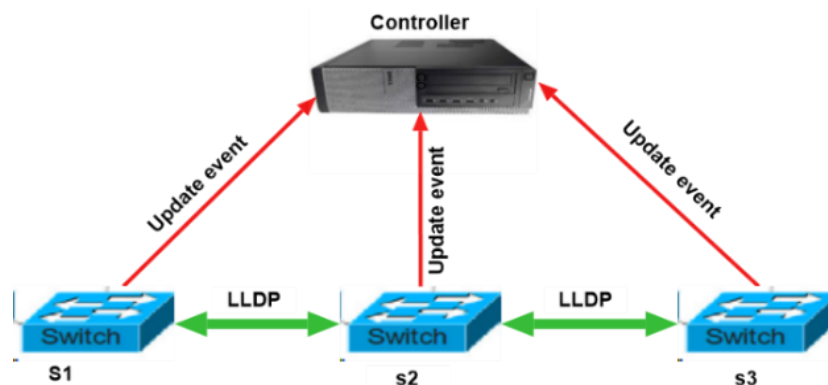


Figure 3. LLDP OF discovery process

For the controller to send and receive LLDP packets to every switch, we have the following activities via pseudo code/algorithm 2.

Algorithm 2.

Input: Generate and process LLDP-packet at the controller

Output: Determine each node location

Initialization:

1. **For** all set of switches in the network **do**
2. **If** Sending LLDP-packet via Packet-Out event to switch a switch from Controller **then**
3. **For** all received LLDP-Packet via packet-in event **do**
4. Insert Link of a switch in the network in all the ports.
5. **End for**
6. **For** all incoming LLDP-Packet **do**
7. **If** LLDP-Packet-In port = controller
8. **Then** Flood LLDP-Packet to all ports except access ports
9. **Else** Send LLDP-Packet to the controller through the packet-in event.
10. **End if**
11. **End for**
12. **End if**
13. **End for**

2.3. Topology DISCOVERY for SDWSN using FTDP

2.3.1. FTDP algorithm packet activities

Figure 4 shows different actions executed upon collecting information in the FTDP protocol. When a packet/information arrives, a certain node executes the following activities via pseudo code/algorithm 3.

Algorithm 3.

Input: All nodes from source to sinks

Output: Determine each node location

Initialization:

1. Receives new TD information.
2. Isolates the header and the payload of the obtained information.
3. **If** the information type is "OPEN PATH", the node learns new rules.
4. **Then** do not drop the packet.
5. **ElseIf** the information sample is "DATA-PACKET",
6. **Then** node A explores the openings of its fuzzy flow table to discover a matching condition/rule.
7. **ElseIf** the requirement is satisfied.
8. **Then** the node runs the better action.
9. **Else** node A transmits a new entry's request from the controller.
10. **If** the sample of a packet is a "BEACON PACKET",
11. **Then** node A updates the new sequence number.
12. **ElseIf** the packet type is "REPORT-PACKET",
13. **Then** Node A transmits the topology information to the closest Sink.
14. **Else** repeat the process.
15. **End if**
16. **End if**

The decision parameters such as RE, and queue length (QL) are presented to the fuzzy framework. The node with the largest NC should be selected.

2.3.2. The fuzzy-logic TD protocol

The purpose of the FTDP is the choice of the best new hop of the decision parameters in the case of the RE, the NC, and the QL in the SDWSN design [28]. In FTDP, the sink gets information about the nodes and decision parameters. At that point, the information is transmitted to the controller. A brief overview of the fuzzy framework is illustrated in Figure 5, showing the three phases: fuzzification, fuzzy inference system, defuzzification.

The membership to a set can either be true or false. For true, there is a member of a set, and false, not a member of a set. In fuzzy systems, the interval for any real value is between 0 and 1 as the degree of membership of a set.

2.3.3. Fuzzification

To measure the node cost (NC) in FTDP, we use a fuzzy system. The NC is determined when using three parameters: the node's RE, queue length, and the number of neighboring nodes. Due to those three parameters, nodes have time-invariant status.

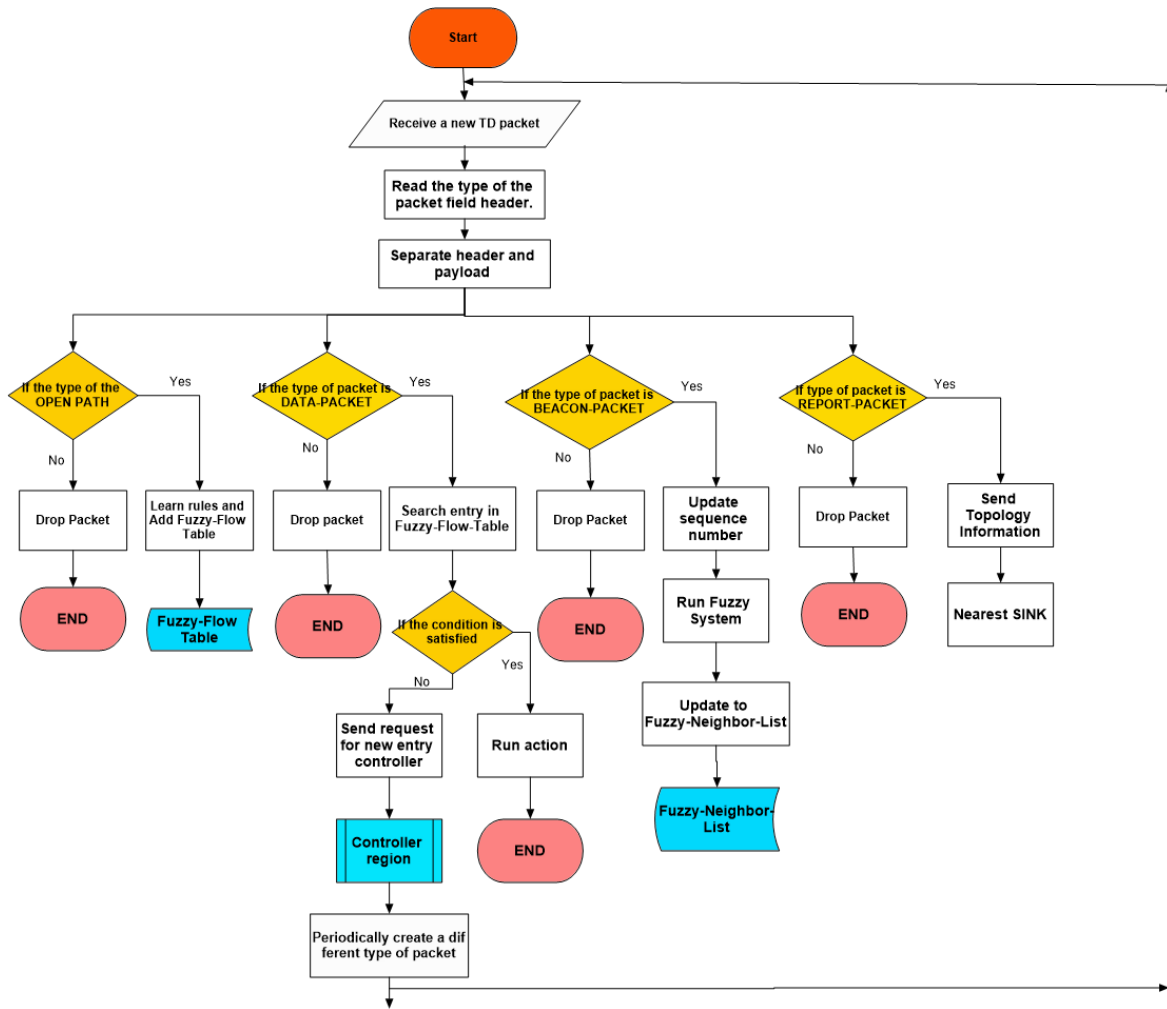


Figure 4. Flowchart of the FTDP

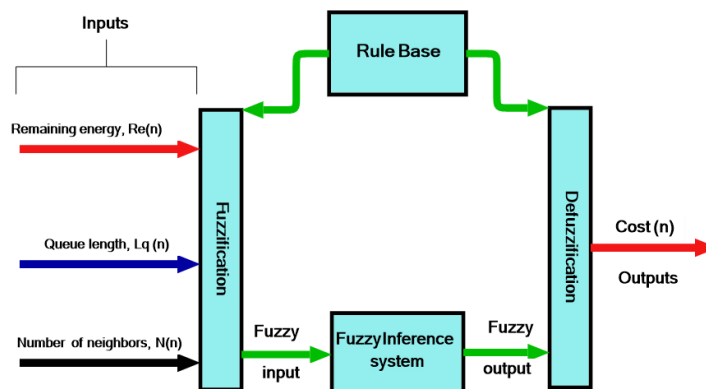


Figure 5. Overview of the fuzzy system

Each graph outlines the input parameter to at least one fuzzy value with their related degrees. The first parameter of each node is the RE. Figure 6 illustrates five phases of the RE: very high, high, medium, low, and very low. For each node we consider the initial amount of RE to be 10 when it is high. For a high value of a node, we need high RE. The node with lower data transmission risk failure is the node that has higher RE.

The next parameter that needs to be considered is the QL of each neighboring node. If the QL is higher, the less it may be chosen as an intermediate node. The last input to consider here in the fuzzy system is the number of neighboring nodes, also called node degree. The higher the degree, the higher the opportunity to be chosen as an intermediate node.

2.3.4. Fuzzy inference system

The fuzzy inference framework is the second stage of the fuzzy framework. This framework gathers all the possible conditions ("if x, at that point y" conditions) that give an inference fuzzy system output to discover the fuzzy conditions that are matching. The fuzzy values are calculated by FTDP in the system called *Mamdani fuzzy inference* [28].

2.3.5. Defuzzification

In this stage, the node cost (NC) is calculated for every node by:

$$NC(n) = \frac{\sum_{i=1}^n Rule_i \times C_i}{\sum_{i=1}^n Rule_i} \tag{1}$$

In (1) shows the NC, where $Rule_i$ is the significance parameter and C_i is the parameter condition. The NC in the network is between 0 and 1 as illustrated in Figure 7. For the node to be taken as the intermediate node, the node with the largest NC must be considered.

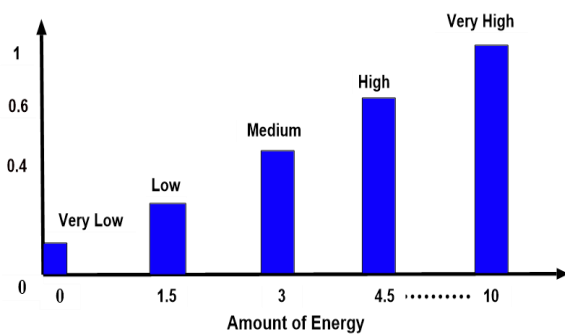


Figure 6. The fuzzy membership capacity of the RE

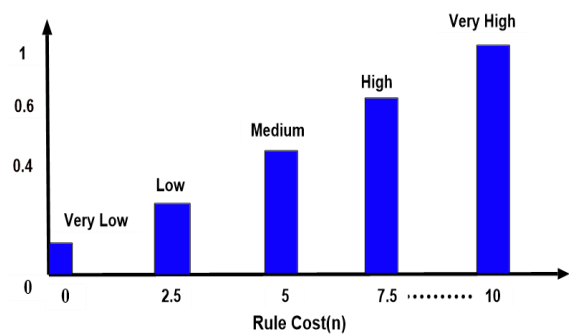


Figure 7. Fuzzy outline for computing the NC of each consolidated circumstance of a node

3. RESEARCH METHOD

Consider node A (with a node ID of 44) that has 5 Joules RE, a QL of 8, and the number of neighboring nodes is 7. The RE of the node is set on low and medium as seen in Figure 8. The vertical line, on the horizontal axis, is in between the low and medium bars. Then on the vertical axis, consider the RE for low to be 0.5 and for medium to be 0.7. For the QL parameter which varies from 0 to 10, let us consider the value of the QL to be 8, then about the vertical line, let us consider the QL medium to be 0.4 and for high to be 0.6.

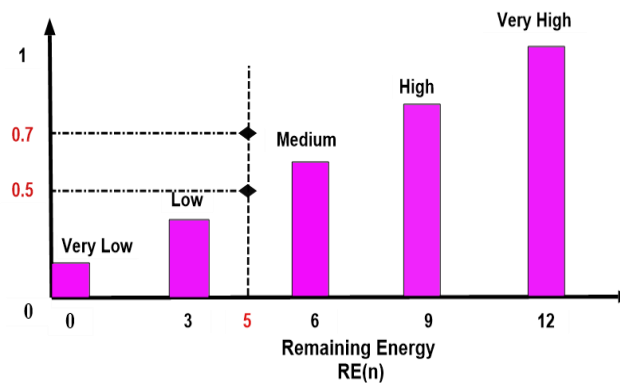


Figure 8. An illustration of the measure of residual energy and two measures of its fuzzy yield

For the number of neighbors, on the vertical line, let us consider the NN for Medium to be 0.8 and for high to be 0.4. To find how many rule states or conditions that one needs to consider, the following equation is used:

$$Rule\ state = \alpha^n \quad (2)$$

where α is the values' number and n the parameters' number.

Thus, for the three parameters, the rule state would be $2^3=8$ states.

The $Rule_i$ is calculated by:

$$Rule_i = RE \times QL \times NN \quad (3)$$

In Table 2, the relating state to each estimation of the three parameters was determined.

Table 2. Results for the rule states

States (<i>i</i>)	Rules	Values	$Rule_i$
1	$RE_{Low} \times QL_{Medium} \times NN_{Medium}$	$0.5 \times 0.4 \times 0.6$	0.12
2	$RE_{Low} \times QL_{High} \times NN_{Medium}$	$0.5 \times 0.6 \times 0.8$	0.24
3	$RE_{Low} \times QL_{Medium} \times NN_{High}$	$0.5 \times 0.4 \times 0.4$	0.08
4	$RE_{Low} \times QL_{High} \times NN_{High}$	$0.5 \times 0.6 \times 0.4$	0.12
5	$RE_{Medium} \times QL_{Medium} \times NN_{Medium}$	$0.7 \times 0.4 \times 0.7$	0.384
6	$RE_{Medium} \times QL_{High} \times NN_{Medium}$	$0.7 \times 0.6 \times 0.8$	0.336
7	$RE_{Medium} \times QL_{Medium} \times NN_{High}$	$0.7 \times 0.4 \times 0.4$	0.112
8	$RE_{Medium} \times QL_{High} \times NN_{High}$	$0.7 \times 0.6 \times 0.4$	0.168

From (1), NC is calculated using the results in Table 2. The selected node will be the one with the highest value of the NC.

$$NC = \frac{(Rule_1 \times C_1) + (Rule_2 \times C_2) + (Rule_3 \times C_3) + (Rule_4 \times C_4)}{Rule_1 + Rule_2 + Rule_3 + Rule_4} + \frac{(Rule_5 \times C_5) + (Rule_6 \times C_6) + (Rule_7 \times C_7) + (Rule_8 \times C_8)}{Rule_5 + Rule_6 + Rule_7 + Rule_8} \quad (4)$$

$$NC = \frac{(0.12 \times 0.5) + (0.24 \times 0.5) + (0.08 \times 0.5) + (0.12 \times 0.05)}{0.12 + 0.24 + 0.08 + 0.12} + \frac{(0.384 \times 0.5) + (0.336 \times 0.5) + (0.112 \times 0.5) + (0.168 \times 0.05)}{0.384 + 0.336 + 0.112 + 0.168}$$

$$NC = \frac{113}{280} + \frac{1061}{2500} = 0.828$$

The above calculations show how to find the node cost (NC) of each neighboring node with different node ID. The selection of the NC is on the node ID with the largest NC.

4. RESULTS AND DISCUSSION

What is new about this paper is the implementation of the FTDP protocol results using the IT-SDN platform, which is an open SDWSN mechanism. It has a simple partition of the protocols used to attain southbound communication (which is the communication between the controller and the data pane) [7], neighbor discovery, and control discovery [37] as shown in Figure 9 on Cooja (Contiki network simulator). Table 3 shows the simulation parameters used in the simulation platform.

The IT-SDN software architecture's components presented in Figure 10, outlines the three main components of the general SDWSN architecture: neighbor discovery (ND) and control discovery (CD) and packet processing.

The number of routing packets, the RE level, and the number of packet losses are the parameters needed for the evaluation of results. A couple of results are defined to examine the performance of this approach in many circumstances. Figure 11 shows the validity of the FTDP protocol for the number of nodes between the sent and received packets. With this result, the ability of the sensor networks deteriorates with the expansion in the network measurement, which is increasing the chance of collisions. In the event of the routing packets' number, the quantity of sent and received information was investigated to determine how efficiently this protocol could transfer packets.

For the next parameter, the information packet delivery rate (PDR), it was defined as the ratio of the received packets/data to the forwarded packets/data.

$$PDR = \frac{\sum Received\ packet}{\sum Sent\ Packets} \times 100 \quad (5)$$

The packet delivery rate is shown in Figure 12 for the validity of FTDP in comparison with SDN-WISE, where the PDR is in percentage and the nodes' number to be from 100 to 500. More nodes led to more multi-hop transmission errors, causing a decrease in PDR.

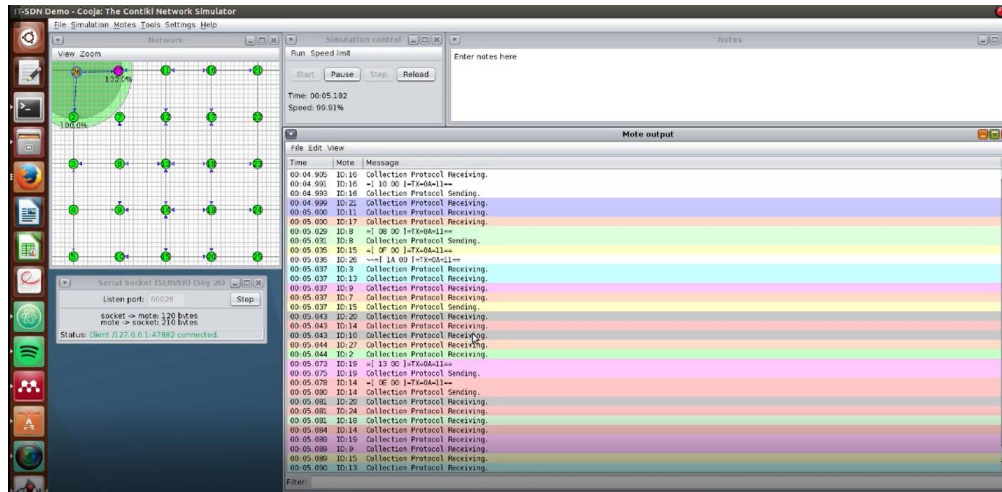


Figure 9. Simulation platform in IT-SDN

Table 3. Simulation parameters

Parameters	Values
Nodes' number	100 to 500
Network size	1000 × 1000
Queue type	Drop tail
Queue size	50
Normal nodes' transmission range	0 to 100 m
Antenna type	Omni-directional
Transmission rate	448 kbps
MAC protocol	802.15.4
Bit rate type	Exponential
Forwarding protocol	UDP
Traffic type	Burst
Deployment strategy	Random
Communication protocol	TCP
Internet Protocol	IPv6

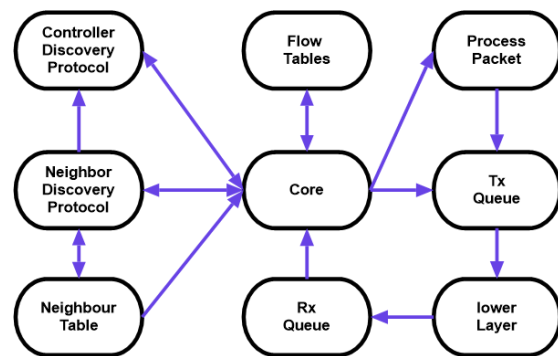


Figure 10. IT-SDN software architecture

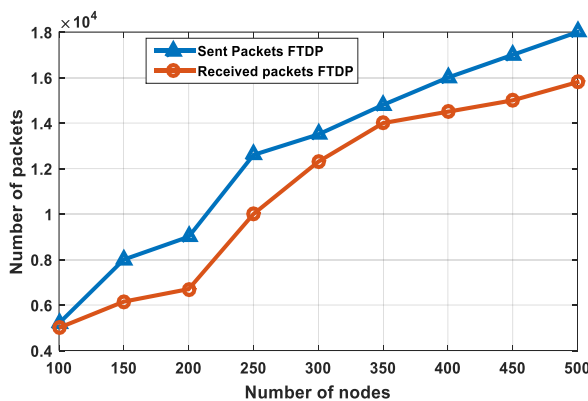


Figure 11. Routing packets for FTDP in the network

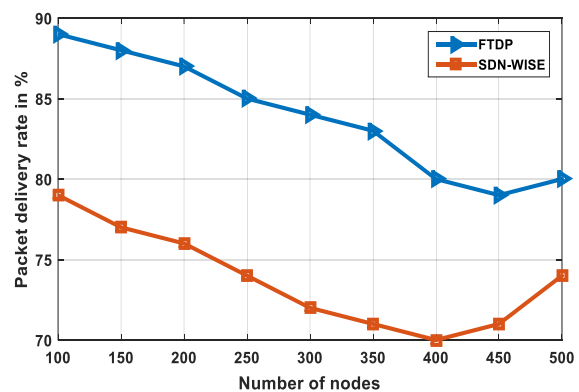


Figure 12. Packet delivery rate in the network

The third parameter that needed to be addressed is the amount of the RE. Energy consumption is one of the serious problems in WSNs. A comparison between the RE of the proposed FTDP and SDN-WISE protocols was made. Figure 13 shows the results of the RE in the network, where the total RE is in Joules for several nodes ranging from 100 to 500. Using FTDP, there is lower energy consumption, leading to an increase in the network total RE.

The last parameter is the lost packets. Figure 14 shows the number of lost packets in the network (in bytes) and the number of nodes from 100 to 500. One of the principal reasons for the number of lost packets in the network is the intermediate nodes' deficiency of energy. From the above results, there is an obvious change in the lost packets' number for FTDP and SDN-WISE protocols.

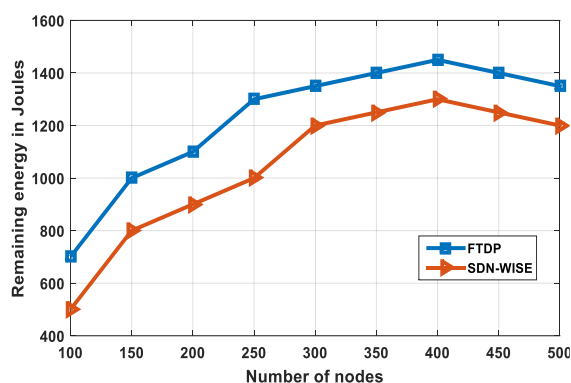


Figure 13. The RE in the network

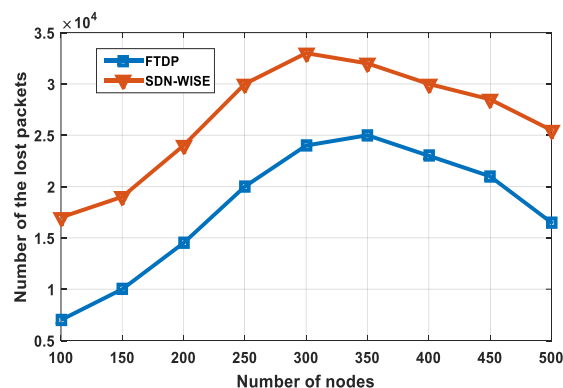


Figure 14. Number of lost packets in the network

5. CONCLUSION

This investigation has addressed the importance of SDN in WSNs. In the study, the transfer of information among the nodes and the master controller plane link with network TD has shown to cause higher energy consumption in the fused SDWDN. A detailed investigation of LLDP-based TD structures in networks made by OF switches was investigated. An improved FTDP protocol using IT-SDN was proposed in the SDN-WISE framework, which supports a fuzzy system in the controller. This is to increase the amount of RE of the network, and by so doing, improving the overall system performance when compared as shown. As future work, the application of FTDP and SDN-WISE framework in a cyber-physical system like a smart grid will be investigated to see how the entire system responds.

ACKNOWLEDGEMENTS

This work was supported by the Global Excellence and Stature under the Fourth Industrial Revolution (GES 4.0) Scholarship, and the Centre of Telecommunications, University of Johannesburg, South Africa.




REFERENCES

- [1] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Computer Networks*, vol. 186, no. December, p. 107739, 2021, doi: 10.1016/j.comnet.2020.107739.
- [2] K. M. Modieginyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey," *Computers & Electrical Engineering*, vol. 0, pp. 1–14, 2017, doi: 10.1016/j.compeleceng.2017.02.026.
- [3] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri and A. Silberschatz, "Topology discovery in heterogeneous IP networks," *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2000, pp. 265-274 vol.1, doi: 10.1109/INFCOM.2000.832196.
- [4] H. Kim and N. Feamster, "Improving network management with software defined networking," in *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114-119, February 2013, doi: 10.1109/MCOM.2013.6461195.
- [5] A. Aissoui, A. Ksentini, A. M. Gueroui and T. Taleb, "Toward Elastic Distributed SDN/NFV Controller for 5G Mobile Cloud Management Systems," in *IEEE Access*, vol. 3, pp. 2055-2064, 2015, doi: 10.1109/ACCESS.2015.2489930.
- [6] S. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani and M. K. Khan, "Topology Discovery in Software Defined Networks: Threats, Taxonomy, and State-of-the-Art," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 303-324, Firstquarter 2017, doi: 10.1109/COMST.2016.2597193.
- [7] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements," *IEEE Access*, vol. 5, no. 12, pp. 1872–1899, 2017, doi: 10.1109/ACCESS.2017.2666200.
- [8] M. Erol-Kantarci and H. T. Mouftah, "Energy-Efficient Information and Communication Infrastructures in the Smart Grid: A Survey on Interactions and Open Issues," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 179-197, Firstquarter 2015, doi: 10.1109/COMST.2014.2341600.
- [9] Y. Wang, H. Chen, X. Wu, and L. Shu, "An energy-efficient SDN based sleep scheduling algorithm for WSNs," *Journal of Network and Computer Applications*, vol. 59, pp. 39–45, 2016, doi: 10.1016/j.jnca.2015.05.002.
- [10] D. Espinel Sarmiento, A. Lebre, L. Nussbaum and A. Chari, "Decentralized SDN Control Plane for a Distributed Cloud-Edge Infrastructure: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 256-281, Firstquarter 2021, doi: 10.1109/COMST.2021.3050297.
- [11] B. Donnet, M. Curie, P. Raoul, L. L. Cnrs, and M. Crovella, "Efficient Algorithms for Large-Scale Topology Discovery," *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, vol. 23, no. 5, pp. 327–338, 2005, doi: 10.1145/1064212.1064256.
- [12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulker, and L. Peterson, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, p. 69, 2008, doi: 10.1145/1355734.1355746.
- [13] N. Kiran, M. S. Kumar, and M. N. Raju, "Improved Step Response of Power System Stabilizer using Fuzzy Logic Controller," *Bulletin of Electrical Engineering and Informatics*, vol. 3, no. 3, pp. 187–194, 2014, doi: 10.12928/eei.v3i3.282.
- [14] G. Tamaras, E. Haleplidis and S. Denazis, "SDN and ForCES based optimal network topology discovery," *Proceedings of the*

- 2015 1st IEEE Conference on Network Softwarization (NetSoft), 2015, pp. 1-6, doi: 10.1109/NETSOFT.2015.7116181.
- [15] J. A. Wickboldt, W. P. De Jesus, P. H. Isolani, C. B. Both, J. Rochol and L. Z. Granville, "Software-defined networking: management requirements and challenges," in *IEEE Communications Magazine*, vol. 53, no. 1, pp. 278-285, January 2015, doi: 10.1109/MCOM.2015.7010546.
- [16] D. Zeng, T. Miyazaki, S. Guo, T. Tsukahara, J. Kitamichi and T. Hayashi, "Evolution of Software-Defined Sensor Networks," *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, 2013, pp. 410-413, doi: 10.1109/MSN.2013.60.
- [17] N. Foster *et al.*, "Languages for software-defined networks," in *IEEE Communications Magazine*, vol. 51, no. 2, pp. 128-134, February 2013, doi: 10.1109/MCOM.2013.6461197.
- [18] C. Cao, L. Luo, Y. Gao, W. Dong and C. Chen, "TinySDM: Software Defined Measurement in Wireless Sensor Networks," *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2016, pp. 1-12, doi: 10.1109/IPSN.2016.7460723.
- [19] B. A. A. Nunes, M. Mendonca, X. -N. Nguyen, K. Obraczka and T. Turtletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617-1634, Third Quarter 2014, doi: 10.1109/SURV.2014.012214.00180.
- [20] P. Jayashree and F. Infant Princy, "Leveraging SDN to conserve energy in WSN-An analysis," *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015, pp. 1-6, doi: 10.1109/ICSCN.2015.7219904.
- [21] T. Luo, H. Tan and T. Q. S. Quek, "Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks," in *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896-1899, November 2012, doi: 10.1109/LCOMM.2012.092812.121712.
- [22] B. Trevizan De Oliveira, L. Batista Gabriel, and C. Borges Margi, "TinySDN: Enabling multiple controllers for software-defined wireless sensor networks," *IEEE Lat. Am. Trans.*, vol. 13, no. 11, pp. 3690-3696, 2015, doi: 10.1109/TLA.2015.7387950.
- [23] S. KhudairLeabi and T. Y. Abdalla, "Energy Efficient Routing Protocol for Maximizing Lifetime in Wireless Sensor Networks using Fuzzy Logic," *International Journal of Computer Applications*, vol. 122, no. 1, pp. 19-25, 2015, doi: 10.5120/21665-4740.
- [24] A. A. Kamil, M. K. Naji, and H. A. Turki, "Design and implementation of grid based clustering in WSN using dynamic sink node," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 2055-2064, 2020, doi: 10.11591/eei.v9i5.1875.
- [25] S. Bera, S. Misra, S. K. Roy and M. S. Obaidat, "Soft-WSN: Software-Defined WSN Management System for IoT Applications," in *IEEE Systems Journal*, vol. 12, no. 3, pp. 2074-2081, Sept. 2018, doi: 10.1109/JSYST.2016.2615761.
- [26] L. Galluccio, S. Milardo, G. Morabito and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless Sensor networks," *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 513-521, doi: 10.1109/INFOCOM.2015.7218418.
- [27] S. Costanzo, L. Galluccio, G. Morabito and S. Palazzo, "Software Defined Wireless Networks: Unbridling SDNs," *2012 European Workshop on Software Defined Networking*, 2012, pp. 1-6, doi: 10.1109/EWSDN.2012.12.
- [28] N. Abdolmaleki, M. Ahmadi, H. T. Malazi, and S. Milardo, "Fuzzy topology discovery protocol for SDN-based wireless sensor networks," *Simulation Modelling Practice and Theory*, vol. 79, pp. 54-68, 2017, doi: 10.1016/j.simpat.2017.09.004.
- [29] A. Banerjee, "Design of a fuzzy-controlled energy - Efficient Multicast Scheduler (FEMS) for SDWSN," *Journal Information Technology Management*, vol. 13, pp. 111-132, 2021, doi: 10.22059/jitm.2021.80758.
- [30] A. Azzouni, R. Boutaba, N. T. M. Trang and G. Pujolle, "sOFTDP: Secure and efficient OpenFlow topology discovery protocol," *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1-7, doi: 10.1109/NOMS.2018.8406229.
- [31] T. Alharbi, M. Portmann and F. Pakzad, "The (in)security of Topology Discovery in Software Defined Networks," *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, 2015, pp. 502-505, doi: 10.1109/LCN.2015.7366363.
- [32] B. Lowekamp, D. O'Hallaron, and T. Gross, "Topology Discovery for Large Ethernet Networks," *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 12, no. 6, pp. 237-248, 2001, doi: 10.1145/383059.383078.
- [33] B. Donnet, F. Timur, and M. Crov, "Improved Algorithms for Network Topology Discovery," *International Workshop on Passive and Active Network Measurement. Springer, Berlin, Heidelberg*, 2005, pp. 149-162.
- [34] R. R. Chpudhury, S. Bandyopadhyay and K. Paul, "A distributed mechanism for topology discovery in ad hoc wireless networks using mobile agents," *2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC (Cat. No.00EX444)*, 2000, pp. 145-146, doi: 10.1109/MOBHOC.2000.869229.
- [35] Y. Bejerano, Y. Breitbart, M. Garofalakis and Rajeev Rastogi, "Physical topology discovery for large multisubnet networks," *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, 2003, pp. 342-352 vol.1, doi: 10.1109/INFCOM.2003.1208686.
- [36] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in OpenFlow-based Software Defined Networks," *Computer Communications*, vol. 77, no. September, pp. 52-61, 2016, doi: 10.1016/j.comcom.2015.09.013.
- [37] R. C. A. Alves, D. A. G. Oliveira, N. S. Gustavo, and C. B. Margi, "IT-SDN: Improved architecture for SDWSN," *In XXXV Brazilian Symposium on Computer Networks and Distributed Systems*, 2017, vol. 8, no. 3, pp. 1-8.

BIOGRAPHIES OF AUTHORS







Joseph Kipongo    received his National Diploma in Electrical Engineering (light current) at the Tshwane University of Technology in 2015, BTech degree in Electrical and Electronic at the same institution in 2017. Currently studying Master of Philosophy (MPhil) in Electrical and Electronics Engineering at the University of Johannesburg, Johannesburg, South Africa, where he was awarded by the University of Johannesburg the Global Excellence and Stature, Fourth Industrial Revolution (GES 4.0) Scholarship in support of his Studies. His research interest are telecommunications, micro-electronics, applied mathematics, digital signal processing (DSP), internet of things (IoT), software defined networks (SDN), machine learning (ML) and digital control systems (DCS). He has been tutoring undergraduate students for more than 5 years in engineering at the University of south Africa and worked in various telecommunication and electronic companies. He is a registered member of IEEE. He can be contacted at email: drebenic4real@gmail.com.



Ebenezer Esenogho     received the Diploma in Computer Engineering (Upper Div.) in 2002/2003, B.Eng. degree (2nd Class Hons Upper-Div.) in Computer Engineering in 2007/2008, the M.Eng. (Upper-Div.) in Electronics/Telecommunication Engineering from the University of Benin in 2011/2012. He previously lectured at the University of Benin. Prior to now, he was involved in research and teaching with the Centre for Radio Access and Rural Technology, University of KwaZulu-Natal, Centre of Excellence, where he rounded-up his Ph.D. degree in electronic (5G Cognitive Network) from 2013-2016, funded by Telkom SA, Eskom, Alcatel, Ericsson, and Huawei. He was hired into the prestigious Global Excellence and Stature (GES) Post-Doctoral Research Fellowship at the University of Johannesburg, Auckland Park under the Institute for Intelligent System (IIS), Center for Telecommunication Research (CfT) to pursue further research. Currently, he doubles as a Senior Researcher with the Center for Telecommunication (CfT) and Research Project Coordinator for Liquid Telecom Research Hub/Centre for Collaborative Digital Networks. His appointment was officially confirmed by the University of Johannesburg as a Senior Research Associate in 2021. Dr Ebenezer is a recipient of several grants/scholarships/awards/fellowships including the CEPS/Eskom's HVDC 2013, CEPS/Eskom's HVDC 2014, J. W Nelson 2015, and GES 2017, 2018, 2019/20. He won the best Oral paper presentation in the postdoctoral conference during his postdoctoral fellowship in 2019. He is passionate about service to humanity and been engaged in several community services both in the Republic of South Africa and his home country. His leadership service is evident as the First Postgraduate Representative for the Student Engineering Council in the Faculty management board of the University of KwaZulu-Natal during his Ph.D. days, 2016-2017. Presently, he served as the first black research representative in the University of Johannesburg Senate/Council, 2018-date. Dr. Ebenezer was a UJ/DST/NRF research delegate to the H2020-ESASTAP EU-South Africa STI Cooperation on Strengthening Technology, Research, and Innovation in Vienna, Austria to woo investors. He has authored/co-authored several peer-reviewed journals and conference papers, chaired sessions in conferences and, reviewed for some notable ISI/Scopus journals in his field. He is very familiar with ISO-9001/18001/14001/22301/27001/20000-1 with a strong affinity for project linked to R&D. His research interests are in but not limited to the fifth generation (5G) wireless networks, cognitive radio networks, network security/cybersecurity, smart grid networks, IoT/IoE, SDN/SDR, wireless sensor networks, artificial intelligence/machine learning, big data & mobile/cloud computing, and visible light communication. He is a registered Engineer in Nigeria and a member of the South Africa Institute of Electrical Engineers (SAIEE) and IEEE Region 8 respectively. He can be contacted at email: ebenezer@uj.ac.za.



Theo G. Swart     was born and educated in South Africa. He received the B. Ing. and M. Ing. Degrees in electric and electronic engineering from the Rand Afrikaans University, South Africa, in 1999 and 2001, respectively (both cum laude), and the D. Ing. Degree from the University of Johannesburg, South Africa, in 2006. He is an associate professor in the Department of Electrical and Electronic Engineering Science and the Director of the UJ Center for Telecommunications. He is a senior member of the IEEE and was previously the chair of the IEEE South Africa Chapter on Information Theory. He is also a specialist editor in Communications and Signal Processing for the SAIEE Africa Research Journal. To date he has over 20 journal articles and over 50 conference articles to his name and has been the co-editor of and contributor to two editions of a comprehensive book on power-line communication. He has served as co-chair for the African Winter School on Information Theory and Communications in 2015 and 2019, technical programme co-chair for the IEEE International Symposium on Power Line Communications and its Applications in 2013, Technical programme track co-chair for IEEE Africon in 2017 and has been a TPC member of various other international conferences. He has done reviews for various journals and conferences, including reviews for the IEEE Transactions on Information Theory, IEEE Transactions on Communications, IEEE International Symposium on Information Theory, IEEE International Symposium on Power Line Communications and its Applications, IEEE Africon and SATNAC. His research interests include information theory, error correction coding, line coding, digital communications and power-line communications. He can be contacted at email: tgswart@uj.ac.za.