Traffic Optimization using Link Weight Routing Algorithm in SDN-based Fog Platforms

MSc Research Project
MSc in Cloud Computing

Shantanu Dileep Paranjpe
Student ID: x19115644

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shantanu Dileep Paranjpe
Student ID:	x19115644
Programme:	MSc in Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Mr. Vikas Sahni
Submission Due Date:	17/08/2020
Project Title:	Traffic Optimization using Link Weight Routing Algorithm in SDN-based Fog Platforms
Word Count:	7262
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	16th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Traffic Optimization using Link Weight Routing Algorithm in SDN-based Fog Platforms

Shantanu Dileep Paranjpe
x19115644

Abstract

Fog Computing is a novel paradigm that assists in overcoming the difficulties related to handling a large amount of sensor data by segregating the applications and services in close proximity to the user. The prime benefits of fog computing are that data is locally stored on the fog nodes instead of transferring the request to the cloud servers which leads to an increase in the latency time. Furthermore, as the data generated by the sensor nodes starts to grow at a significant rate, it leads to network congestion, and to control the network packet loss and network congestion, effective techniques are required to transmit the data to the destination node with minimum transmission overhead and latency. In this project, Software Defined Networks have been integrated into the wireless fog topology generated in the Network Simulator 2. The SDN Controller gets initialized at the beginning of the simulation where it gathers the data related to the network links and nodes. In addition to this, link weight routing algorithm is executed by the SDN controller. The algorithm instructs the Fog SDN controller to choose the best path among all the network links based on the signal strength of the links. The SDN controller assigns a weight using this algorithm and routes the traffic using the best path towards the destination. The performance of our proposed methodology is analyzed by altering the count of nodes in the topology. It also helps in resolving the limitations of IoT and helps in minimizing the transmission overhead and routing delay in the Fog based IoT networks with the use of the customized routing algorithm.

1 Introduction

The Internet of Things is an evolving technology that comprises of an intelligent ecosystem involving heterogeneous devices and technologies as illustrated by Tayyaba et al. (2017). A typical Fog IoT network consists of devices like RFID reader devices associated with a tag, wireless communication devices, actuators, and other wireless devices like wireless sensors. These devices connect to global networks like the Internet and this leads to the formation of the IoT network. The devices that are part of the Fog IoT networks are heterogeneous and a variety of applications like machine to machine, vehicular ad-hoc networks, wireless sensor networks, and cellular networks are deployed as part of the IoT networks. It is a complex process to handle the heterogeneous data generated by the wireless IoT devices. The Fog-enabled Internet of things adheres to a layered design that contains three layers namely the Device layer, Network layer, and the application layer. The device layer mainly contains the wireless sensing devices and objects, the responsibility of the network layer is to transfer the data from the wireless objects to the

Fog devices like the RFID readers, gateway devices like the router and other devices like a switch. As per the report generated by Cisco, there will be a rise of IoT devices from the current count of 6 billion to 50 million IoT devices in 2020.

The devices that are part of the IoT networks produce a large amount of data that is difficult to handle. Numerous solutions have been built to handle the diverse data generated by wireless devices. The conventional IoT network is not capable of managing a large number of devices getting connected to the Internet daily. So, Software Defined Networks have been used in the IoT networks to handle the ubiquitous data generated by the wireless sensor. The Software-Defined networks overcome the challenges posed by the traditional networks and have proved to be beneficial in the management of IoT devices with the use of data and control planes as presented by Braun and Menth (2014). The Software Defined Networks segregate the control plane and data plane. The control plane which is considered as the primary component takes the routing decisions and creates the flow entries and the data plane comprise the data forwarding devices like switches and routers that transfer the data from source to the destination. The optimization of routing efficiency is an unaddressed area of the Internet of Things. As the number of IoT devices getting to the Internet is increasing at an alarming rate, an enormous quantity of data is created in the Fog IoT networks. As the traditional networks are not able to handle the huge volume of data, it leads to packet loss and network congestion and also increases the average latency of the IoT networks. SDN is a novel paradigm of networking that has divulged its utility in minimizing the complexities in the management of the IoT networks as stated by Gonzalez et al. (2016). SDN consists of a centralized or distributed SDN controller that provides a universal view of the entire network and this helps in taking the decisions related to the management of IoT devices in a much faster way. This project implements a link weight routing algorithm which is executed by the OpenFlow FogSDN Controller to assign weights to the network links on the basis of the signal strength. A combination of SDN and fog computing can assist in resolving issues like high latency and transmission overhead in the IoT networks.

1.1 Motivation and Background

In the initial days of the Internet, the use of the Internet was limited to transferring the data packets between the data sources and the users allocated a particular IP address. Due to the growth of Internet technology, it is now being used for sharing of data among various low power constraint devices which are smaller in size and when interconnected in millions, form a network known as the Internet of Things. An IoT or Internet of Things is a collection of devices that are heterogeneous and are connected to the Internet. These devices perform the task of sensing the environment around them like detecting the temperature, humidity, and send the sensing data to the Internet for further analysis. As per the analysis made by Cisco, the count of IoT devices will grow up to 125 million till 2030 with an annual increase of 12%. As per the report generated by Cisco for measuring the current IoT growth, 6.4 million IoT devices are connected to the Internet and will increase up to 50 billion in 2020. These IoT devices create a large amount of data such as data generated in the year 2018 is around 6.2 Exabyte which has grown up to 30.6 Exabyte in 2020. Multiple solutions have been developed for effectively managing the IoT data. However, the conventional network is not proficient in handling such a vast amount of data. To overcome this challenge, Software Defined Networks has proved to be a promising technology for managing the heterogeneous IoT network with the use of programmable control and data planes. The IoT and SDN Integration can assist in controlling and handling the IoT data in varied scenarios.

In a Fog environment, different types of IoT devices have different configuration and setup. IoT consists of four Cs namely Compute, Create, Connect, and Collect to handle the revolution of IoT. The number of IoT devices connected to the Internet is growing in large numbers. A huge volume of traffic is generated through the IoT devices and imposes a heavy load on the IoT sensor network. If the data generated through these devices is not routed efficiently, it may lead to packet loss and minimize the IoT network performance and also increase the overall routing delay. To overcome these issues, there is a need of an efficient routing mechanism that can seamlessly handle the IoT traffic.

1.2 Research Question

Research Question: *"Can routing efficiency in SDN-based Fog platforms be optimized using Link Weight Routing Algorithm"?*

2 Related Work

The routing algorithms play a vital role in routing the IoT traffic as the sensor data produced by the sensor devices is sensitive to delays and has to be route in a fast and intelligent manner. The important parameters to be considered while routing the IoT data is the routing delay, transmission overhead and throughput. The below section provides an overview of all the major routing protocols used in IoT platforms.

2.1 Routing Protocols used in IoT

The Ant Colony routing algorithm has been proposed by Said (2017) for optimizing the selection of the appropriate path for routing the data in the IoT network. It bifurcates the IoT environment into classified areas depending on the type of network. Once it is done with the bifurcation process, it utilizes the most appropriate ant colony algorithm to the concerned IoT network within each area. The ant colony routing algorithm offers competitive performance in terms of bandwidth consumption, routing delay, throughput, and packet loss ratio.

The author Chen et al. (2012), proposed a concept of a new routing protocol that can be used in the wireless sensor networks. The protocol is known by the name Context-Aware Sea Computing routing protocol. This protocol takes into consideration the main feature of the Internet of Things called Context-Awareness. It routes the data as per the location of the device. Depending on the model used for Sea computing, the author tries to modify the conventional routing protocol by making use of the Markov model for probability and intelligence theory for artificial intelligence theory which consists of directive and forwarding rules used for routing the packet.

In order to overcome issues like load balancing, energy efficiency, and reliability, an extended version of routing protocol based on link quality and energy efficiency was proposed for Wireless Sensor Networks and IoT by Machado et al. (2013). The REL protocol proves to be beneficial in improving the data quality, transmitting the IoT data with low latency, minimizing the packet drops, and improving the overall reliability of data. In REL, a mechanism based on events is used to achieve the load balancing in order to improve the performance of the system. It also helps in moving the energy from one node to other neighbouring nodes with the use of an on-demand system. The Energy-efficient content routing protocol was introduced by Chelloug et al. (2015) in order to reduce the overall energy consumption in sensor devices in IoT. The algorithm creates a topology that can be used in a centralized manner for routing the analytical data generated by the sensor devices to the intended recipient nodes connected to the Internet. The underlying characteristics of energy efficient EECBR can be attained by building a virtual topology that helps in balancing the energy load amongst IoT devices with the help of the self-organization property of the obtained hierarchy.

da Silva Fré et al. (2015) suggested a distributed CCR method that considers the traffic mitigation accomplished by the aggregation of content-centered data while routing traffic over secure communication links by integrating connection quality information. On the basis of the contents of the message, a separate routing is created by each of the sensor nodes by executing the unique objective function. The prime objective is to route the content which is heterogeneous in nature through selected communication links to those nodes which have the capability of processing and extracting accurate information. This helps in minimizing the redundant traffic communication and also lowers the overall transmission count of the packets that travel from the sensor devices.

Hu et al. (2014) developed an immune orthogonal learning PSO algorithm which serves as a faster route recovery from the path failures due to agility and flexibility of the sink node and also stores the information related to the alternative path through which the sensor data can be routed. The results clearly demonstrate that the algorithm helps in reducing the overhead incurred in the communication among the sensor nodes and also enhances the network lifetime. da Silva Fré et al. (2015) made use of the Particle Swarm Optimization (PSO) algorithm to measure the various levels of the transmission power needed by each of the sensor nodes without creating detached areas in the cluster. The final simulation results indicated that more energy can be saved from sensors by using PSO compared to common node deployment with sole emitting power.

Carvalho et al. (2016) proposed a novel algorithm known as the Robust Shortest Path Tree(RSPT) which helps in enhancing the overall strength of network routing by taking into account the weakness present in the quality of the link and to tackle the price of individual links with the assistance of a single problem value. Verma et al. (2017) developed a Computational Intelligence to preserve device and energy resources by migrating the CI tasks from the IoT appliances to the cloud storage for further analysis.

Shailendra et al. (2017) utilized LTE technology to deliver coverage for several IoT devices with an intention of making these devices to act in a controlled manner and to enable a transmission in an efficient manner using the LTE Random Access Channel (RACH) mechanism.

2.2 Routing Protocols used in Wireless Sensor Networks

2.2.1 Location-based Routing Protocols

In the routing protocols based on the location, the routing path is chosen by checking the location of the node as an alternative to checking the routing tables. This expands the network life cycle and helps in curtailing the range of transmission for the nodes.

Karim and Nasser (2012) developed a location-based and fault-tolerant clustering protocol for mobile WSN to augment the overall consumption of energy and decrease the end-to-end delay incurred in the transmission of packets from the sensor nodes. A unique fault-tolerant feature was clubbed with the routing protocols to detect the failures in the sensor networks and the data links. But the simulation results indicated that the protocol did not provide good performance in terms of minimizing the end-end delay.

Ukani and Thacker (2015) by lifting the energy-sensing mechanism. The writer mainly centered on the Quality of Service (QoS) to extract information related to the n-hop neighborhood, which fulfilled the requirement of the network intangible environments like video surveillance. The simulation outcome revealed that the anticipated protocol was able to take a routing decision in a more efficient way than the greedy forwarding. However, additional parameters should be considered in the simulation for accurate results.

Sun et al. (2017) presented a routing protocol known as Speed Up-Greedy Stateless Routing protocol based on position in WSN. The key idea was to merge the greedy mode with the speed-up mode to deliver a solution for increasing the data transmission and for reducing the routing delay. Likewise, the planned next-hop selection could be utilized to both mobile and still nodes by contemplating the energy consumption. Ultimately, the performance parameters like the packet delivery ratio, hop count end-to-end delay were thoroughly tested in a simulated environment and the outcome displayed an improvement in the overall performance of the wireless sensor networks.

2.2.2 Hierarchical Routing protocols

In the hierarchical routing protocol, the network is bifurcated into numerous clusters. Each of the clusters is owned by a CH node for controlling the communication between the nodes. The data is transferred to the gateway node across the aggregated convergence which helps in decreasing the traffic communication and preserve the energy usage among the sensor nodes.

Joshi et al. (2016) proposed a new framework for Wireless sensor networks. It is used for processing the sensory data generated by the sensor devices. The writers used a protocol known as the Low Energy Adaptive Clustering and the simulation results showed that the protocol helped in minimizing the count of packets and the overhead in transmitting the packets from the sensor nodes to the base station. The benefit of using this protocol is traffic monitoring, compression, and decompression and filtering, where is easily integrated into the intended framework. One of the limitations of this protocol is that its performance was never tested in the production environment.

2.2.3 Data-oriented Routing Protocols

Data-based routing mechanisms are taken into consideration for data fusion problems. Node cooperation is used to improve the efficiency of transmission and energy savings of the network.
Samaras and Triantari (2016) explored the Direct Diffusion Routing Algorithm (DDRA) in a detailed manner. This routing algorithm is based on the clustering selection system and direct diffusion to identify the network topology and energy consumption of the sensor nodes. After the evaluation of the protocol results proved that the DDRA could resolve the issue related to network delay and rate of packet delivery in an IoT environment.

A protocol known as SPIN was introduced by Grover et al. (2014) by making use of a mechanism called as selective forwarding to boost the performance in data delivery. In the event some of the packets were unrecognized by the source node, the source node would repeat the task of forwarding them to the destination node. Jain and Khan (2014) broadly examined the two leading data-centric protocols namely SPIN and DIFF protocols. With use of Tool Command Language, Network Animator and NS3 simulator respectively, their performance was measured based on the control overhead, PDR, throughput, and end-to-end delay. The simulation test results showed that the performance of DDIFF protocol is better than the SPIN protocol under the mobile and static conditions.

2.3 SDN and IoT Integration

The Software-Defined Network and the Internet of things are two emergent technologies. The count of IoT devices connected to the Internet is increasing by leaps and bounds and IoT data management is a complicated task for a big distributed network. The main objective of IoT is to connect the smart devices to the Internet and when combined with Software Defined networks facilitates in managing the network with more ease by bifurcating the control and data plane. SDN manages the IoT network in a programmatic way and also provides flexibility in handling the enormous data generated by the sensor devices.

The first plan for a wireless network centered on SDN cellular network was developed by LI LE and Rexford (2012). In this architecture, policies are devised for all the users based on attributes that can be used and supported in the LTE network and also achieve a fine-grained control in the IoT network. In order to overcome the issue of complexity in the SDN cellular networks, a novel SDN framework was developed by S. Tomovic. et al. in Gudipati et al. (2013). A centralized SDN controller was used to isolate the entire network into a separate geographical domain. A centralized control plane isolates the whole RAN into a geographical domain. A large base station with a centralized controller provides allocation of resources in a framework consisting of three elements namely, and frequency, time, and space slots. This framework, when tested in the cellular SDN network, proved beneficial in decreasing the overall delay.

In order to manage the WSN network, Galluccio et al. (2015) recommended the use of SDN-Wise framework, where SDN-based IoT data aggregation module and duty cycle is offered as a state-full SDN solution. The SDN-Wise translates the data between the

WISE-Visor and the sensor node. SDN-Wise frameworks consist of application programming interface for creation of SDN controller in all the supported programming languages in networks like OMNet++ network. Lantz et al. (2010) identifies mechanisms for the handling and definition of the Open-Flow rules which makes SDN-Wise a stateful framework as opposed to the conventional OpenFlow networks which are stateless in nature. Qin et al. (2014) augmented the concept of Multi-network SDN controller system for handling the IoT heterogenous networks in a campus area. MINA is a middleware based on principles like self-healing, monitoring and flexibility concerning the changes in the network. It is developed in an SDN style layered framework and its primary function is to match the flows sent to it by the forwarding plane devices thereby the gap in the semantics of IoT in multi-network system.

Zhang and Li (2014) performed the analysis and evaluation of RPL protocol by using a simulator known as Contiki Cooja Simulator. The evaluation of the overall performance was done during the operating stage of the Wireless sensor networks. The performance of the RPL (Routing Protocol for low power and lossy networks) was assessed based on the performance metrics like convergence time, signalling overhead and latency of the message. The test results revealed that the latency for transferring a particular packet does not increase as per the total load of packets. The reason for this is that with lower number of packets, the nodes cut the radio communication for preserving the energy. The other parameters like signalling overhead and convergence need improvement in terms of the performance.

Moreover, as WSN networks are inherently data-centric, Qin et al. (2014) proposed data-centric routing algorithms that create awareness among the network protocols concerning the contents of a packet. Consequently, SDN-WISE can manage packets on basis of the content stored in their payload and header fields. Packets in the Open-Flow supported network are categorized based on similarity that exists in a particular field in the header of the packet. SDN-WISE also supports grouping can be accomplished centered on more intricate relational operators, e.g. lower than, higher than, distinct from, etc.

Tayyaba et al. (2017) recommended a framework for SDN based IoT control and management. In this framework, data is gathered from the sensor board and accumulated on the IoT Bridge. The assembled data is forwarded to the SDSec controller for purpose of checking the security. Access is granted to only authorized devices by using authorization and authentication. Rules are created on the forwarding devices by the SDN controller for routing the data to the appropriate destination and managing the policies related to the monitoring of data as per the data accumulated data by the SDN controller and collected into the SDStore module of the framework.

3 Methodology

This section provides a summary of the tools, techniques, and the algorithm to be used in our project.

3.1 Network Simulator 2

Network simulator is also known as the NS2 is a widely used network simulation tool that has proved its utility in the study of the dynamic behaviour of interaction and enterprise networks. NS2 makes use of a script interpreter known as the OTCL (Object-oriented TCL) that comprises of network object component libraries and an event simulator that is executed in the backend along with the combination of the plumbing module libraries. The NS2 simulator offers users the ability to control and manage the behaviour of these routing algorithms in real-time. The NS2 simulator simulates the network protocols such as the UDP and TCP, traffic patterns like the Telnet, Constant Bit Rate (CBR), and routing protocols like Dynamic Source Routing and Ad-hoc Distance vector Routing. In the NS2 simulation process, the user runs the OTCL scripts combined with the ns command which start the initialization of the event scheduler and also helps in setting up the network topology with the use of plumbing modules and the network objects in the library and thereby providing the information to the application agents to start and stop the transmission of packets. The Network animator window is displayed after the execution of the ns command for depicting the simulation scenarios. The workflow of the NS2 simulation is consists of the below sequence of activities:

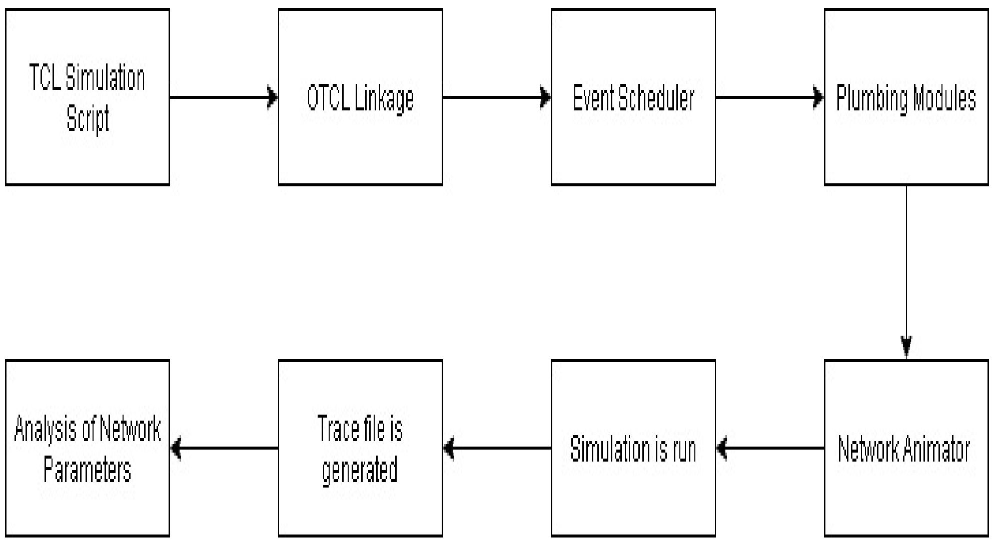

Figure 1: NS2 Simulation Workflow

3.2 Open-Flow SDN Controller

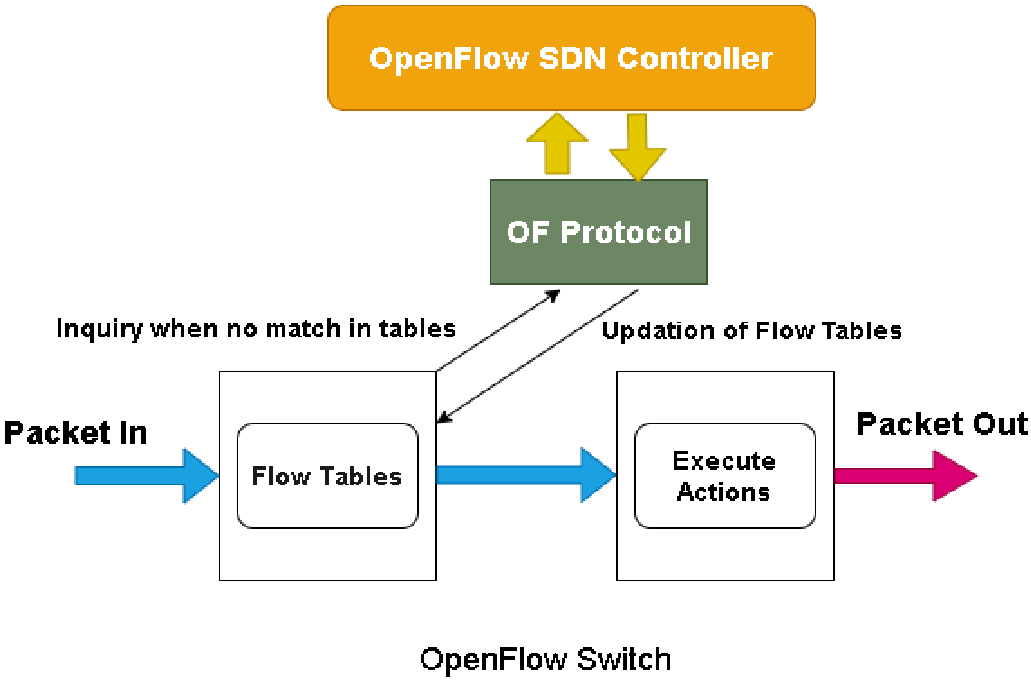

Figure 2: Open-Flow SDN Controller Framework

The Open-Flow SDN Controller has been implemented within the project so on enable the network administrator to dump the control plane of all the switches by connecting to a central controller for outlining the behavior of the network. The OpenFlow controller can be programmed in such a way that it can control the data-forwarding rules for all the forwarding devices with the use of the OpenFlow protocol. When the SDN-Switch receives the request, it will check if the flow rules are available for the IoT packet. If the matching flow rule is not available in the switch, the IoT request will be forwarded to FogSDN Controller. The controller will check the network graph and will send the updated flow table entries to the SDN switch which will use the forwarding devices to transfer the IoT request to the appropriate destination.

3.3 Link Weight Routing Algorithm

The link weight routing algorithm is executed by the SDN controller. The SDN controller uses this algorithm to compute the appropriate network path to be taken by the IoT packets to reach the destination. This helps in reducing the overall delay and also helps in preventing network congestion in the IoT network. The link weight routing algorithm takes the source node and the destination node along with the network graph as the input. Then it starts the process of discovering the nodes present in the graph and updates the network graph with the node information. It also finds the nodes located at the edge of the graph called as the edge nodes. It starts an iterative process of determining the weights of the network links based on the signal strength. After the finding the bandwidth of all the network paths, it generates the route vectors to ascertain the route that needs to be taken by the IoT packets.

4 Design Specification

This section provides a brief overview of the design workflow of our project and also provides a summary of the set of activities and technologies that are part of the architectural framework.

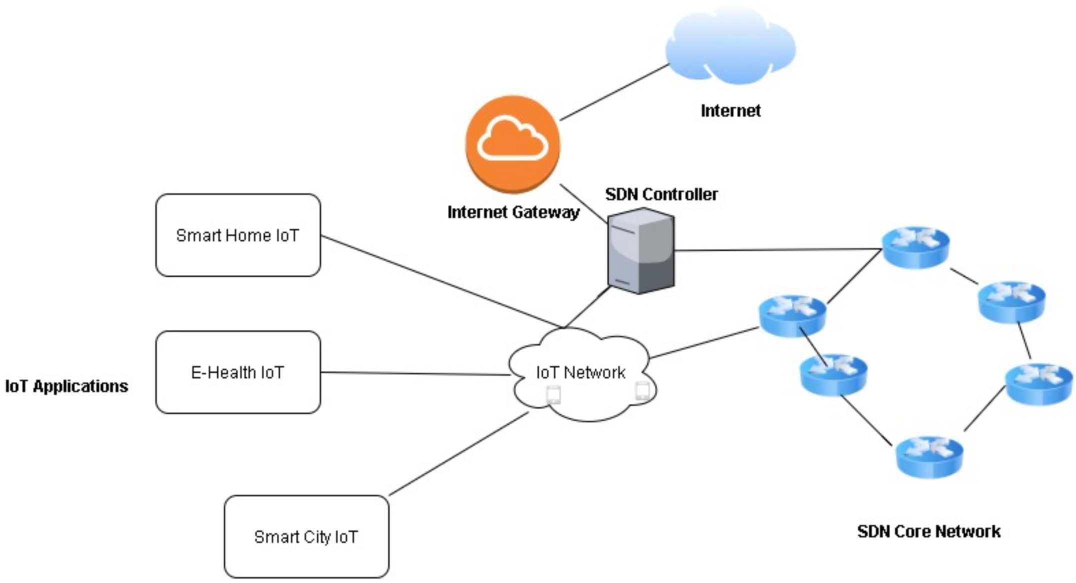

Figure 3: SDN based IoT Architecture

As observed in the figure, the infrastructure layer comprises of the IoT sensors that generate the IoT traffic and transfer it to the IoT controller. The IoT controller manages the traffic generated by the IoT sensors. All the IoT sensors are connected to the fog devices present at the fog layer. The fog devices collect the data sent by the IoT controller and sends it to the Fog SDN controller which acts as a central orchestration manager. The control plane in SDN is a physical server that is running the network orchestration software to control and manage the network and the data plane comprises the data forwarding devices like router and switches to transmit the data from the wireless sensor nodes to the destination sensor node. The control plane and the data plane communicate with each other through the OpenFlow protocol. The OpenFlow SDN Controller has been used along with the Link Weight Routing algorithm which will run inside the SDN controller so as to identify the appropriate network path to forward the data. The fog SDN controller takes care of controlling infrastructure and application layers. Link weight routing algorithm is proposed for effective traffic management using the multi-layer SDN architecture in fog IoT platforms. The temporary data stored by the fog devices is passed to the Fog gateway. The Fog gateway connects the fog devices to the Internet where the data is analysed and stored for longer duration. The link weight routing algorithm takes the source node and the destination node along with the network graph as the input. Then it starts the process of discovering the nodes present in the graph and updates the network graph with the node information. It also finds the nodes located at the edge of the graph called as the edge nodes. It starts an iterative process of determining the weights of the network links. After finding the bandwidth of all the network paths, it generates the route vectors to ascertain the route that needs to be taken by the IoT packets on the basis of the link parameters and chooses the best path towards the destination.

4.1 Sequence Diagram

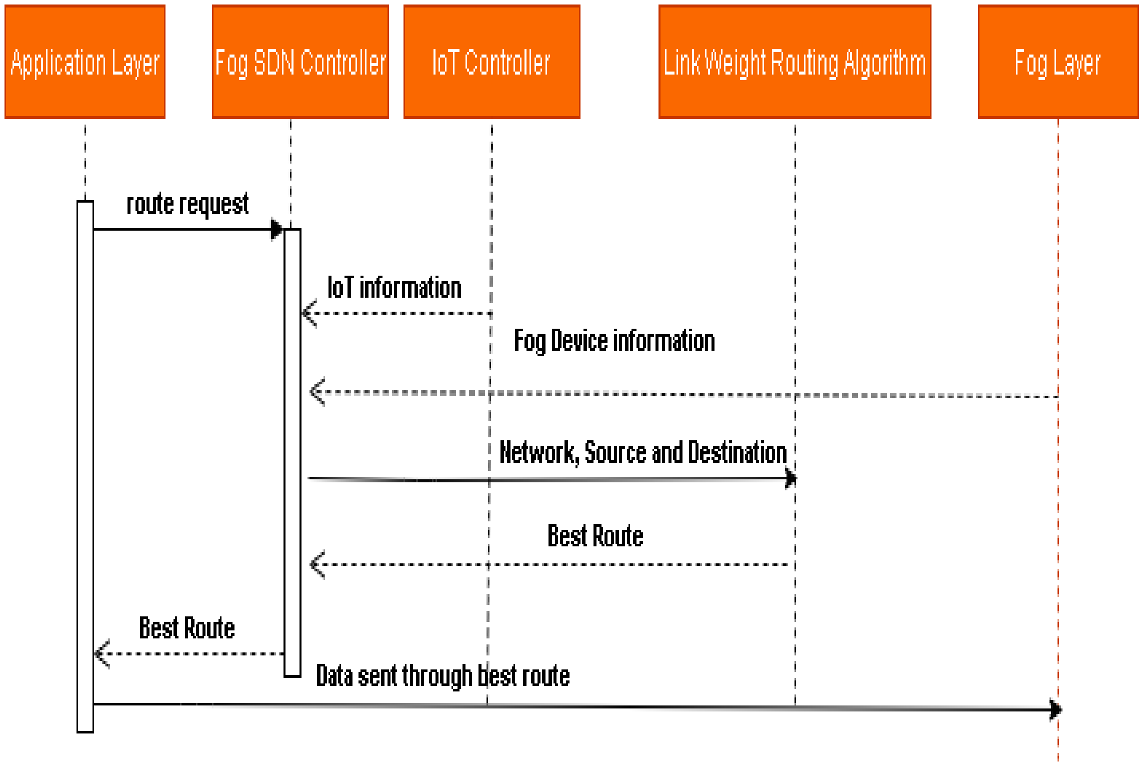

Figure 4: Sequence Diagram

The sequence diagram illustrates the sequence of activities that are carried out by different components of the topology. As seen in the diagram, the application layer consists of the IoT applications running on the IoT infrastructure layer. These applications continuously exchange data with each other. Latency is one of the prominent issues when the services are executed on the cloud infrastructure. To overcome this limitation, the services are implemented in the fog layer to minimize the latency and improve the overall throughput and response time. In the proposed topology, the services running on the application layer send a route request to the Fog SDN controller to route the service request to the appropriate destination node. The Fog SDN controller sends the network graph information, the source and destination node information as input to the Link weight Algorithm. The Link weight routing algorithm takes decision about the best path the packets can take based on the signal strength, energy levels and the reliability of the links. It sends the best path information to the Fog SDN controller which sends the data to the services present in the application layer. The application layer transmits the IoT service packet to the services executing on the destination fog node. So, to conclude these are the sequence of activities that take place between the service running on the source node and the destination server.

5 Implementation

This section provides the description of the implementation model and the tools used for the developing the implementation prototype.

5.1 NS2 Wireless Topology

The wireless topology in this project has been created in Network Simulator 2. The simulator has been installed on a virtual machine created in Oracle VirtualBox with Ubuntu guest operating system installed on it. The network topology mainly comprises of 28 wireless nodes and two sink nodes. In this topology, the Fog SDN Controller is initialized at the beginning of the simulation. The controller builds a network graph consisting of all the network links and nodes. It invokes the link weight routing algorithm to determine the best path from the source node towards the destination node and forwards this data to the SDN Switch and the switch stores this data in the flow table and passes the best path data to the RFID reader and sensor nodes. The IoT device layer consists of wireless sensor devices associated with an RFID tag. The fog layer consists of RFID readers which act as fog devices. The wireless sensor devices send the security credentials along with the IoT traffic to the RFID reader node. The reader node receives the packet and sends it to the SDN switch. The switch provides the best path data to the RFID reader which forwards it to the IoT device. The source IoT node in our topology is node number 2. The node 2 forwards the data towards the destination using the best path information. The proposed topology helps in optimizing the traffic management and also reduce the overall routing delay in the IoT sensor networks as compared to conventional routing algorithms.

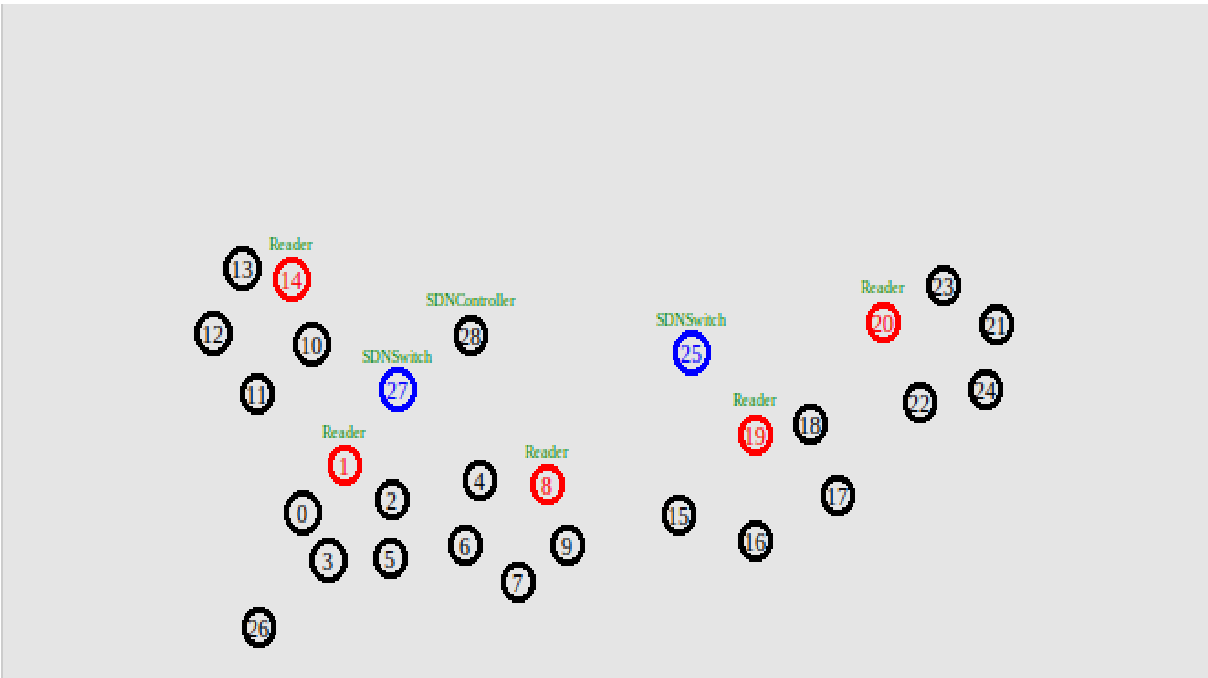

Figure 5: NS2 Wireless Topology

5.2 Link Weight Routing Algorithm

The objective of the link weight routing algorithm is to determine the best path among all the available paths in the network graph and route the IoT application services data with maximum throughput and reduced response time. It starts with the initialization process where it builds a graph of the entire network, then identifies the total number of edges connected to each of the wireless nodes. The network node information is stored in variable N, E variable is used to store the information related to the network edges and the route information is stored in the R. It starts with the discovery process where all the nodes present in the are identified and accordingly the network graph is updated. Furthermore, for all the nodes n in the network, it assigns an appropriate weight to the wireless link based on the metrics like the signal strength and energy levels of the link. The link with the higher strength and energy is allocated the highest weight and likewise, the algorithm allocates weights to all the links in the network and accordingly updates the route vector matrix. For all the routes in the route vector, the algorithm ascertains whether the current route is better than the previous route and identifies the best path for sending the IoT service packets to the destination node. The pseudo code for the link weight algorithm is as follows:

Algorithm 1 Link Weight Routing Algorithm

```
Input: Network graph G, source S, destination D.
Output: Efficient traffic management
 Initialize network vector N.
 Initialize edge nodes vector E.
 Initialize route vector R.
 Initialize all routes vector A.
 N=DiscoverNodes()
 G=UpdateGraph(N)
 E = FindEdgeNodes(G)
 for each node n in N do
    Associate weight to the link.
    Update R.
    Add R to A.
 end for
 for each route a in A do
    Check with the previous route.
    if  current route better than previous route then
       Track the best route.
       Return the best route.
    end if
 end for
 =0
```

6 Evaluation

This section provides description of the results and analysis of the experiments performed in the existing and proposed strategies using the NS2 simulation scenarios.
In order to evaluate the performance statistics, 2 experiments were performed by creating the wireless network topology in NS2 simulator. In the first experiment, the performance of the existing routing algorithms is analyzed and in the second experiment, the performance of the proposed algorithm is analyzed by integrating SDN functionality in the network topology.

6.1 Experiment on the existing AODV and DSDV Routing Algorithms without SDN

In the first experiment, wireless topology has been built for AODV and DSDV routing algorithms using the scenario generator software. In this experiment, 5 iterations were performed and in each iteration, the count of nodes was incremented by 10, starting from 10 to 50. The output of the experiment was captured in the simulation trace file. For evaluating the data in the trace file, awk scripts were used for gathering the data related to performance metrics like Delay Time and the Throughput. The results were plotted in a graphical format using the values captured in the trace file.

6.1.1 Average end-end Delay

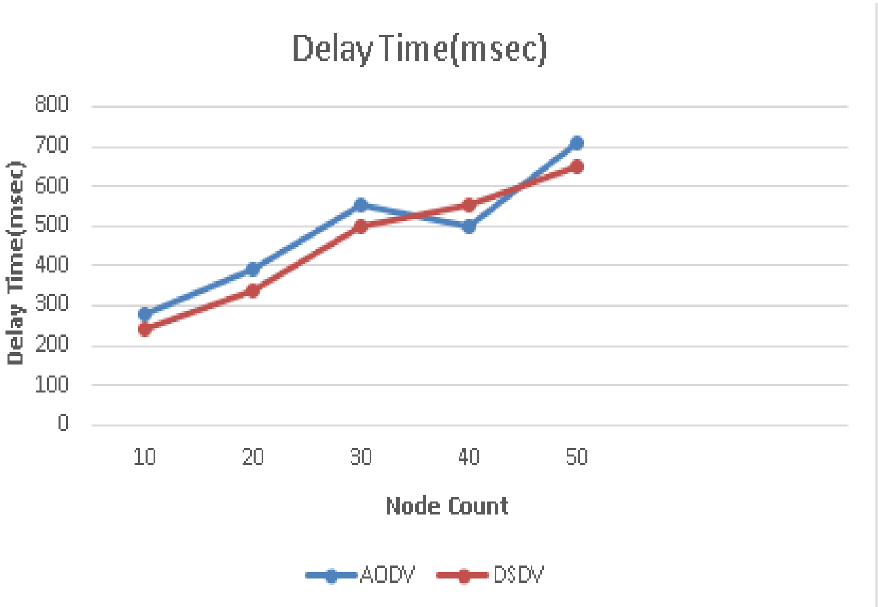

Figure 6: Average end-to-end Delay

The average end-to-end delay is calculated as the difference in time at which the sender sent the packets to the time at which the destination node received the packets. For calculating the Delay time, an awk script was used. The X axis displays the Node Count and the Y-axis denotes the Delay time in milliseconds. As it is clearly visible in the graph, the average delay time for the AODV algorithm is higher than the DSDV algorithm.

6.1.2 Throughput

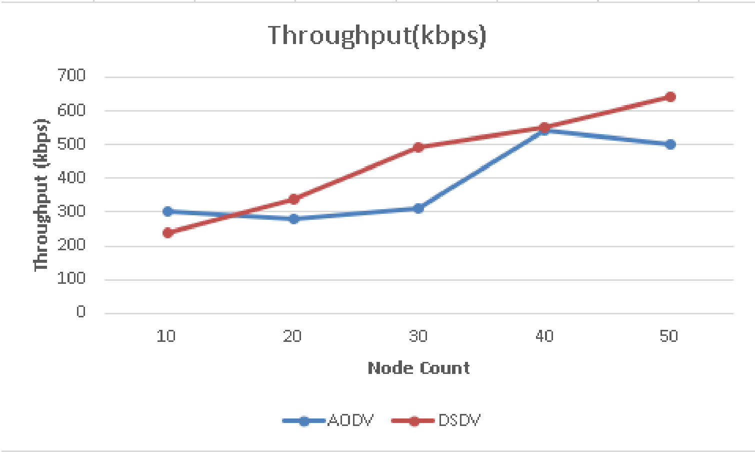

Figure 7: Throughput

The throughput is calculated as the rate at which the packets were sent from the source to the destination and is measured in kb/s. Throughput is calculated with use of an awk script. The X axis displays the Node Count and the Y-axis denotes the Throughput in kb/s. The graph clearly indicates that the DSDV algorithm offers higher throughput than that of the AODV algorithm.

6.2 Experiment on the Proposed Link Weight Routing Algorithm using SDN

In the second experiment, a wireless topology is created for testing the proposed methodology using the link weight algorithm. The experiment was carried out on total 50 wireless nodes where total 5 iterations and count of nodes was increased by 10 in each of the iterations. The output was captured in the simulation trace file. For evaluating the data in the trace file, awk performance scripts have been used. The results were plotted in a graphical format.

6.2.1 Average End-to-End Delay

The Delay Time of the proposed algorithm is measured in milliseconds. The X axis displays the total number of nodes and the Y-axis denotes the Delay Time in milliseconds. The graph clearly indicates that our proposed algorithm outperforms the existing algorithms and provides lower delay time.

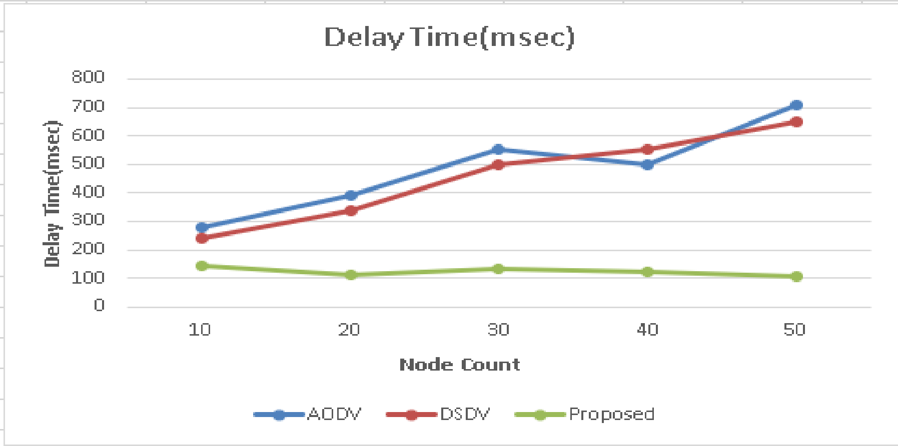

Figure 8: Average End-to End Delay

6.2.2 Throughput

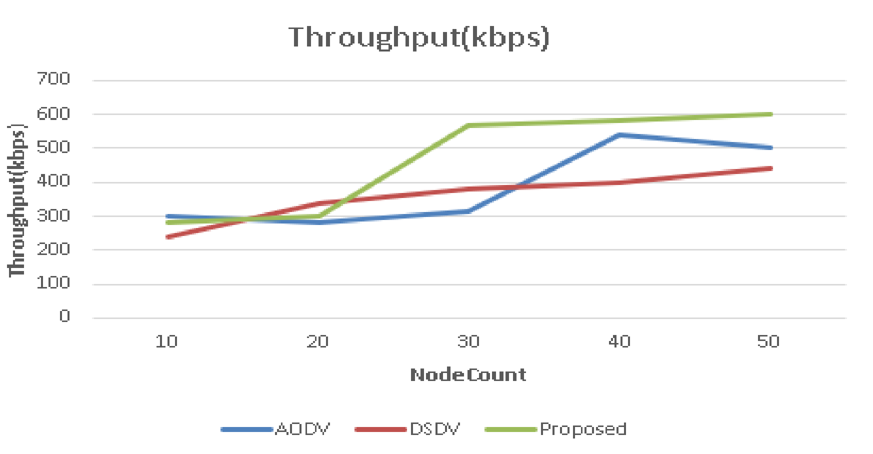

Figure 9: Throughput

Throughput of the proposed algorithm is measured in kbps. The X axis displays the total count of the nodes and the Y-axis denotes the Throughput in kb/s. The graph clearly indicates that our proposed algorithm provides better throughput when compared with the existing AODV and DSDV algorithms.

6.3 Discussion

This research paper helps in addressing the limitations in the Fog IoT environment by using a customized algorithm demonstrated in Network Simulator 2 by creating a wireless topology. The SDN Controller executes a customized routing algorithm known as the Link weight routing algorithm. The algorithm is executed at the beginning of the simulation by the controller to assign weights to the links based on the energy level and the signal strength. The controller routes the packet by choosing the best path to the destination. The proposed methodology has been evaluated by performing 2 experiments, where the 1st experiment is related to the performance analysis of the existing routing algorithms. The outcome of the 1st experiment is that DSDV algorithm provides better results than AODV and in the 2nd experiment, the performance of the existing algorithms has been evaluated by integrating SDN and on comparing the results of existing and proposed methodology, it is clearly evident that the proposed algorithm provides better performance as compared to the existing algorithms used in NS2.

7 Conclusion and Future Work

The prime objective of this research paper is to tackle major issues like routing delay, transmission overhead and low throughput by integrating SDN within the IoT fog layer. The performance of the proposed algorithm is analyzed by using 2 sets of experiments, where the 1st experiment is related to the performance analysis of the existing routing algorithms. The outcome of the 1st experiment is that the DSDV algorithm provides higher packet delivery, high throughput, and lower delay as compared to AODV and in the 2nd experiment, the customized algorithm has been evaluated by making use of awk performance scripts. On comparing the results, it is observed that the proposed algorithm provides better throughput and packet delivery ratio and minimizes the routing delay.

The existing setup can be extended in the future by deploying the proposed methodology in an enterprise-level environment and assessing the response time, throughput, and delay by using a real-world IoT application on top of it. In addition to this, distributed fog SDN architecture can be operated where multiple SDN controllers can be used to segregate the IoT requests to balance the load among all the controllers. Furthermore, the fog SDN controller can be connected to a global cloud controller located in the cloud layer to forward requests for which flow rule is not available in the flow table of the fog SDN switch. The cloud controller will have a global view of the entire network and the local cloud controller can be programmed in such a way that it gives priority to the latency-sensitive data and rest of the data for which immediate response is not required can be sent to the cloud controller which would then store it temporarily in the cloud servers. This architecture can help to further reduce the overall delay and increase the throughput and response time as there the traffic will be managed in a programmatic manner using the software-defined networks at the fog layer and the cloud layer.

References

Braun, W. and Menth, M. (2014). Software-defined networking using openflow: Protocols, applications and architectural design choices, *Future Internet* **6**(2): 302–336.

Carvalho, I. A., Noronha, T. F., Duhamel, C. and Vieira, L. F. (2016). A scenario based heuristic for the robust shortest path tree problem, *IFAC-PapersOnLine* **49**(12): 443–448.

Chelloug, S. A. et al. (2015). Energy-efficient content-based routing in internet of things, *Journal of Computer and Communications* **3**(12): 9.

Chen, Z., Wang, H., Liu, Y., Bu, F. and Wei, Z. (2012). A context-aware routing protocol on internet of things based on sea computing model, *Journal of Computers* **7**(1): 96–105.

da Silva Fré, G. L., de Carvalho Silva, J., Reis, F. A. and Mendes, L. D. P. (2015). Particle swarm optimization implementation for minimal transmission power providing a fully-connected cluster for the internet of things, *2015 International Workshop on Telecommunications (IWT)*, IEEE, pp. 1–7.

Galluccio, L., Milardo, S., Morabito, G. and Palazzo, S. (2015). Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks, *2015 IEEE Conference on Computer Communications (INFOCOM)*, IEEE, pp. 513–521.

Gonzalez, C., Charfadine, S. M., Flauzac, O. and Nolot, F. (2016). Sdn-based security framework for the iot in distributed grid, *2016 international multidisciplinary conference on computer and energy science (SpliTech)*, IEEE, pp. 1–5.

Grover, J., Sharma, M. et al. (2014). Reliable spin in wireless sensor network, *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, IEEE, pp. 1–6.

Gudipati, A., Perry, D., Li, L. E. and Katti, S. (2013). Softran: Software defined radio access network, *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pp. 25–30.

Hu, Y., Ding, Y., Hao, K., Ren, L. and Han, H. (2014). An immune orthogonal learning particle swarm optimisation algorithm for routing recovery of wireless sensor networks with mobile sink, *International Journal of Systems Science* **45**(3): 337–350.

Jain, V. and Khan, N. A. (2014). Simulation analysis of directed diffusion and spin routing protocol in wireless sensor network, *2014 Conference on IT in Business, Industry and Government (CSIBIG)*, IEEE, pp. 1–6.

Joshi, J., Awasthi, P., Mukherjee, S., Kumar, R., Kurian, D. S. and Deka, M. J. (2016). Seed: Secure and energy efficient data transmission in wireless sensor networks, *2016 4th International Conference on Information and Communication Technology (ICoICT)*, IEEE, pp. 1–6.

Karim, L. and Nasser, N. (2012). Reliable location-aware routing protocol for mobile wireless sensor network, *IET communications* **6**(14): 2149–2158.

Lantz, B., Heller, B. and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks, *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pp. 1–6.

LI LE, M. and Rexford, J. (2012). Toward software-defined cellular networks [c], software defined networking (ewsdn), *2012 European Workshop on, pp7-12*.

Machado, K., Rosário, D., Cerqueira, E., Loureiro, A. A., Neto, A. and De Souza, J. N. (2013). A routing protocol based on energy and link quality for internet of things applications, *sensors* **13**(2): 1942–1964.

Qin, Z., Denker, G., Giannelli, C., Bellavista, P. and Venkatasubramanian, N. (2014). A software defined networking architecture for the internet-of-things, *2014 IEEE network operations and management symposium (NOMS)*, IEEE, pp. 1–9.

Said, O. (2017). Analysis, design and simulation of internet of things routing algorithm based on ant colony optimization, *International Journal of Communication Systems* **30**(8): e3174.

Samaras, N. S. and Triantari, F. S. (2016). On direct diffusion routing for wireless sensor networks, *2016 Advances in Wireless and Optical Communications (RTUWO)*, IEEE, pp. 89–94.

Shailendra, S., Rao, A., Panigrahi, B., Rath, H. K. and Simha, A. (2017). Power efficient rach mechanism for dense iot deployment, *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, pp. 373–378.

Sun, Y., Guo, J. and Yao, Y. (2017). Speed up-greedy perimeter stateless routing protocol for wireless sensor networks (su-gpsr), *2017 IEEE 18th International Conference on High Performance Switching and Routing (HPSR)*, IEEE, pp. 1–6.

Tayyaba, S. K., Shah, M. A., Khan, O. A. and Ahmed, A. W. (2017). Software defined network (sdn) based internet of things (iot) a road ahead, *Proceedings of the International Conference on Future Networks and Distributed Systems*, pp. 1–8.

Ukani, V. and Thacker, D. (2015). Qos aware geographic routing protocol for multimedia transmission in wireless sensor network, *2015 5th Nirma University International Conference on Engineering (NUiCONE)*, IEEE, pp. 1–6.

Verma, A., Kaushal, S. and Sangaiah, A. K. (2017). Computational intelligence based heuristic approach for maximizing energy efficiency in internet of things, *Intelligent decision support systems for sustainable computing*, Springer, pp. 53–76.

Zhang, T. and Li, X. (2014). Evaluating and analyzing the performance of rpl in contiki, *Proceedings of the first international workshop on Mobile sensing, computing and communication*, pp. 19–24.