Energy Efficient Routing for Internet-of-Things Based Agriculture in Rural Areas

Mohammed Khomeini Bin Abu[1,*], Paulson Eberechukwu Numan[1], Kamaludin Mohamad Yusof[1], Mohamad Kamal Bin A. Rahim[1], Mohamad Rijal Bin Hamid[1], and Shaik Mazhar Hussain[2,*]

[1] Faculty of Electrical Engineering, University Teknologi Malaysia, Malaysia
[2] Department of Computing and Electronics, Middle East College, Muscat, Oman
Email: mkhomeini2@graduate.utm.my (M.K.B.A.); enpaulson2@hanyang.ac.kr (P.E.N.); kamalmy@utm.my (K.M.Y.); mdkamal@utm.my (M.K.B.A.R.); rijal@utm.my (M.R.B.H.); mazhar@mec.edu.om (S.M.H.)
*Corresponding author

***Abstract*—In the last few years, IoT routing has been a daunting job for researchers. This is a difficulty because traditional routing algorithms are not appropriate for IoT-based sensor systems. Near the sink, the nodes absorbed more energy and died. This leads the network to be split, which limits the network's existence. The sensor node is mainly constrained by energy. The sensors are electronic equipment operated by the battery. In certain systems, it is impossible to substitute the batteries. The greater packet transfer would bring more tension to the battery-powered nodes, and the entire existence of the network would be impacted. IoT networks have no defined and extremely complex infrastructures. The complex infrastructure is responsible for storage; the sensor nodes have insufficient battery energy. Algorithms for energy-conscious routing have demonstrated a significant decrease in electricity use. IoT thus requires a protocol to route effectively. This research proposes an original efficient routing mechanism for IoT networks. The aim of the research lies in designing a modified Open Shortest Path First (OSPF) routing algorithm that is energy efficient. We have modified OSPF to give preference to healthier paths based on the criteria of the total energy available on the path, the path length, and the avoidance of critical nodes. The proposed methodology was evaluated through extensive simulations conducted in a real-world setting on the Johor Bahru UTM campus, covering an area of approximately 1 square kilometer with a roughly rectangular shape. The results demonstrated significant improvements in energy efficiency and network longevity.**

***Keywords*—Industry 4.0 (IR4.0), Internet of Things (IoT), Smart agriculture, rural network, sensor network**

I. Introduction

The fourth industrial revolution, also referred to as Industry 4.0 (IR4.0) is a famous topic today as most of the researchers and industrial players discuss it. Technology Categories for IR4.0 are divided into numerous categories [1], and The Internet of Things (IoT) is one of them. The IoT represents a network that provides communication between "things" (i.e., objects or devices) by using sensors via information and communication technology infrastructure [2], which results in real-time sensing and actuating abilities. IoT turns conventional artefacts into autonomous intelligent objects in key technology such as the sensor network, pervasive computation, enhanced knowledge, increased actions, rules, protocols, or embedded systems. Today, IoT is one of the major subjects toward a developed country. The main benefits for a company that's applying IoT system are productivity improvement, predictive analysis, rapid response, and reduction of human error. In the coming decades, agriculture will become more important than ever. The UN aims to hit the 9.7 billion people of the world by 2050, with a 69% increase in the world's agriculture production by 2050 [3, 4]. Farmers are moving to the Internet of Things (IoT) to satisfy their needs for analytics and improved processing capability. IoT applications (agricultural, transport, car surveillance, industrial, commercial applications, construction, residence, temperature, and city monitoring) are widespread [5, 6] as shown in Fig. 1.

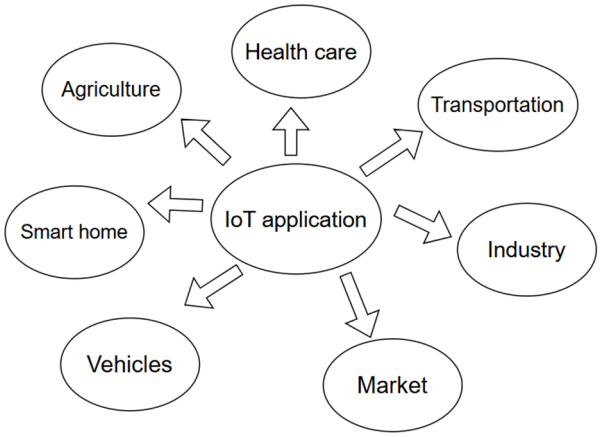


Fig. 1. IoT application [2].

The IoT forms and easily builds on the future of computers and computer networks [7]. The entire concept arose from the ubiquitous nature of several internet-connected artefacts or objects. These devices with a range of on-board sensors are becoming inexpensive and

Manuscript received April 15, 2024; revised May 25, 2024; accepted June 14, 2024; published August 20, 2024.

widespread and are becoming powerful and smaller. Many IoT implementations have stringent needs, such as dense network, low cost, energy-efficient, unattended operation, ease of sensor node placement, low maintenance costs, slimming, and so on. The IoT will bring the future of agriculture to the next level. Smart agriculture or digital farming is now becoming more widespread among farmers, and the usage of agricultural drones and sensors soon becomes the norm for high-tech agriculture. IoT applications in farming have been illustrated in Fig. 2 and how farmers are to satisfy global food demands over the next few years through "IoT farming". Given the advantages of IoT in agriculture, it is a very difficult challenge to provide rural areas with last-mile internet connectivity, that is, several promising attempts have been made through the years [8, 9]. Rural internet technology must overcome obstacles such as power, capital, and infrastructural shortages respectively.

The rural areas need some form of network and internet connectivity that may help drive IR4.0 and smart agriculture [10, 11]. Farmers may use networked devices to track crops and livestock, or also to remotely run individual machinery, saving farmers time and pain to look at the land in person or manually manage all equipment. Setbacks to networking in rural areas are distance to the city, high cost of infrastructure, and poverty level of the users. Although the Malaysian government plans to improve the network coverage as shown in Fig. 3, National Digital Infrastructure (JENDELA) Lab Report on 3 September 2020, the telecommunication company (telco) is unwilling to invest due to no incentive for low population area. Many urban residents enjoy quality and self-recognized wired internet connections [12].

However, for some, fast quality, stable broadband is a privilege that only rural regions would think about. Simple internet service is mostly, at best very poor and pricey and at worst non-existent. Problems in the construction and management of long wire distances amongst scattered rural communities render wired links unattractive. In comparison, the large distances to be travelled by wires to render remote regions open for these networks prohibit internet service companies from serving certain rural areas with the same services [13].

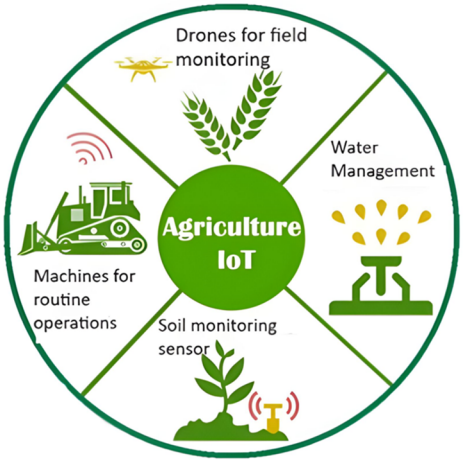


Fig. 2. IoT Applications in agriculture [5].

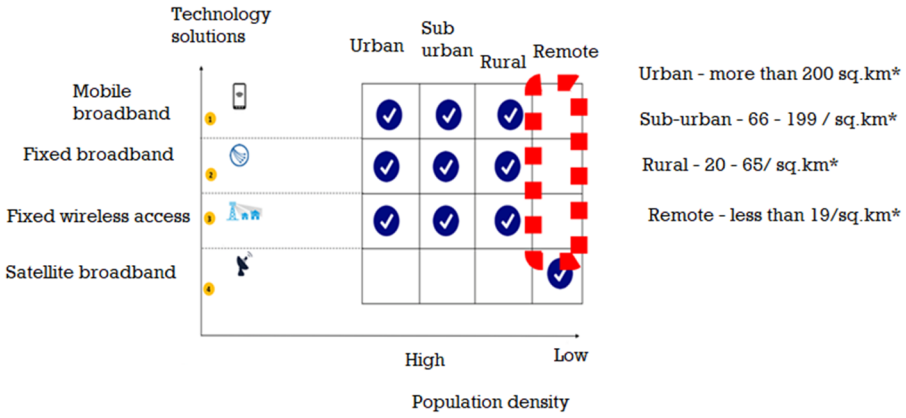


Fig. 3. Technological solutions vs population density [2].

The usage of "fixed" internet links in cabling and optical fiber may be a first option alternative. However, the approach is expensive and calls for the tenants to be next to the exchange points physically [14]. For this cause, Internet Service Providers (ISP) must invest in infrastructures, including long lines of control, repairers, key switches, interchange switches, and other components in many rural areas and developing countries [15]. To attain a strong return on investment, the move must be in heavily developed regions, such as medium-sized cities, etc. Nevertheless, there is a small community in several remote regions and low wages, i.e., a digital divide. A comparative study has been done on the Wireless Mesh Network, Wifi Based Long Distance (WiLD), and Satellite technology solutions for rural internet access [16]. Wireless Broadband networks are an answer to the high costs and drawbacks of stretching cables for broad rural residents. The Wi-Fi solution enables low-cost connectivity to the whole network. These towers are technically fitted with numerous entry points, solar, and rechargeable batteries. Through building these towers in the centres of the cities, several citizens will share them at the same time.

While wireless alternatives in rural areas have all potential benefits, many rural areas also have no wireless connectivity [17]. Sometimes this is due entirely to limitations of technology. Residences can be tens of miles apart in rural areas, making it difficult to propagate a signal to many homes around a base station. There are typically

around 2,000 people per square mile in urban areas versus 10 in some rural areas. The fact that homes are so sparse in rural areas also hinders companies from making the investments necessary to supply rural areas with quality Internet service, even when the technology exists to make such service possible. Also, the wireless solution has some disadvantages in the form of the limited range the WiFi standards (802.11a/b/g/n/ac) offer and the form of the limited transmission power available in the node return path. A more beneficial solution is to provide a passive repeater, which consists of two connected passive antennas. One antenna is placed outside the edge of the coverage area pointing at the WiFi base station, and the other antenna is placed inside the WSN. Nevertheless, this solution has a limitation; since sensor nodes are driven by the battery and in many applications, these batteries cannot be replaced. They die when the battery exhaust and the network functionality are affected. Thus, a routing technique is very much essential to enhance the life span and manage the battery efficiently. This characteristic motivates us to design energy-efficient routing techniques. This paper addresses some of the problems involved in the IoT for agriculture. We have proposed an IoT network as the backbone to the sensor network for agriculture. The involvement of IoT in agriculture is not very prominent due to the lack of technological awareness of the farmers and the cost of the IoT devices. Most of the rural farmers are poor, and cost plays crucial in the selection of technology for agriculture for them. Taking this cost factor and low power requirement of the IoT devices, we have proposed a low-cost IoT network for smart agriculture.

There are various IoT sensors for measuring different parameters for the actualization of smart agriculture using IoT. Every sensor is fitted with a sensor chip, a radio transfer module, a power supply, a data processing, and storage devices within an IoT-based sensor network [18]. These modules are therefore constructed with immense resource constraints such as resources and processing power. The biggest disadvantage in IoT is the sensor power supply limitation. The sensor node is mainly constrained by energy. The sensors are electronic equipment operated by the battery. Also, it is not possible to change the batteries of thousands of sensors because of the solitary and aggressive existence of the sensing environment. The greater packet transfer would bring more strain on the battery-powered nodes, and the entire existence of the network would be impacted. IoT networks have no defined and extremely complex infrastructures. The complex infrastructure is responsible for storage; the sensor nodes have insufficient battery energy. The energy efficiency of sensors therefore presents a challenge that requires study to extend the existence of the network from months to years [19].

Also, IoT-based systems require a long-living network. Clustering and routing are the most employed methods for effective energy utilization and scalability control of the network to maximize the lifetime of the device. It allows for power saving in sensors as a certain aggregation mechanism allows cluster heads to delete redundant and false-positive data. This retains the bandwidth of contact. The inter-cluster routing may also be handled properly since only APs retain the routing details against the Base Station (BS) [20]. Several sensor network architectures allow sensors in a multi-hop routing to send collected data to a mobile or static BS [21, 22]. A stable and low-power data collection and routing system to BS remains a challenge, however.

Several works have undoubtedly been researched to develop IoT and sensor design power control techniques [23–27]. These studies cover low power usage and computers, Medium Access Control (MAC), clock synchronization strategies, and protected data aggregation. The IoT network is an intermediate-node multi-hop network that transmits data. There is a strong probability of failure across nodes. The level of link loss impacts the data transmission ratio directly and lowers network efficiency. The produced data should enter the Access Point (AP) in several applications as soon as possible. However, routing direction, router position, and router failure frequency are not available to boost the end-to-end latency [28, 29].

Algorithms for energy-conscious routing have demonstrated a significant decrease in electricity use. IoT thus requires a protocol to route effectively. Due to dense deployment, redundant data are generated by the sensor nodes, and several copies of the same data are accessible to the base station. IoT routing has been a daunting job for researchers. This is difficult because traditional routing algorithms are not appropriate for IoT-based sensor systems, such as those used in IP-based networks, as such conventional routing algorithms use tables of routing. IoT's intrinsic features do not equate to the traditional wireless networks [30–34]. It's a very complex and application-specific network and has minimal energy, storage, and workmanship. These features find designing a routing protocol quite difficult.

A. Wireless Communication Technologies for Agriculture

Information of the IoT technologies used in different agricultural applications like hardware systems and wireless communications technology will be addressed in this portion. The various wireless protocols and specifications used in agriculture are addressed in this portion. Different IoT cloud service providers, which are common for these applications in the current sector, are also being examined. In this section, we are introducing a survey of the moving technologies that are either the most significant elements of our study or the IoT. Technologies that enable the interaction layer in the IoT to work. We direct you to documents such as [34–38] and references to achieve a more in-depth knowledge of a broader spectrum of IoT technologies. The two metrics are obstacles for the current agriculture approach [5, 8] in the contrast of these wireless systems, which are still the strongest technology in power usage and connectivity. IoT is a revolution for the future realm that links anything that can use a link. In the field of IoT, cellular technologies are being extended and developed [9]. The latest IoT-System Narrowband-IoT (NB-IoT) is developed from existing Long-Term Evolution (LTE) features. After this, NB-IoT will segment the LTE spectrum without issues of coexistence, and use the same pieces of equipment and seamlessly link to the main LTE

network. This enables the help of all network facilities such as surveillance, monitoring, regulation, loading, and authentication. NB-IoT's architecture targets involve a higher coverage spectrum, longer battery life (i.e., ten years), higher network size (52,000 devices/channel/cell), and affordable devices. But soon NB-IoT innovations including Long-Range Radio (LoRa) are to be used because of low power usage and ideally for the broadcasting of agricultural knowledge over long distances.

B. Relationship between IoT Agriculture and Emerging Technologies 6G / AI in agriculture

The relationship between IoT, 6G, and AI in agriculture is multifaceted and synergetic. Enhancing efficiency, productivity, and sustainability. Here are a few key points highlighting their interconnection:

(1) IoT for Data Collection: IoT devices like soil moisture sensors, weather stations, and GPS-enabled machinery collect real-time data on crop conditions, soil health, and environmental factors. Wearable IoT devices for livestock track health metrics, location, and behaviour, ensuring better management of animal welfare [39].
(2) 6G for Enhance Connectivity: 6G networks will provide ultra-low latency and high-speed data transfer, crucial for real-time monitoring and decision-making in agriculture. Enhanced 6G coverage will bring robust internet access to remote and rural farming areas, enabling widespread adoption of smart farming technologies [40].
(3) AI for Data Analysis and Decision Making: AI algorithms analyze the data collected by IoT devices to predict weather patterns, pest infections, and crop diseases, helping farmers take preventive measures. AI-powered robots and drones can perform tasks like planting, weeding, and harvesting with high precision, reducing labour costs, and increasing efficiency.
(4) Integration of IoT, 6G, and AI: Combining IoT sensors, 6G connectivity, and AI analytics creates a comprehensive smart farming system where data is seamlessly collected, transmitted, and analyzed in real-time, allowing for immediate responses to changing conditions.

The adoption of Internet of Things (IoT) technology in the agricultural sector, especially in rural areas, has shown significant potential for improving productivity, resource management, and sustainability [35]. IoT-based sensor networks enable real-time monitoring of soil conditions, crop health, weather patterns, and other critical parameters, facilitating data-driven decision-making. Several studies have explored IoT routing in rural agricultural settings.

This research focused on deploying IoT sensor networks to monitor soil moisture levels in rural farms. The study utilized traditional routing protocols and highlighted the challenges of maintaining network connectivity and energy efficiency in large, sparse deployments.

Developed a customized routing protocol for IoT networks in agricultural fields. Their protocol aimed to optimize data transmission based on environmental conditions and node energy levels. While effective, the protocol was limited by its complexity and high computational requirements, making it less suitable for resource-constrained sensor nodes.

Proposed an adaptive routing algorithm designed for crop monitoring systems in rural areas. The algorithm adjusted routes based on network traffic and node that the algorithm's performance degraded under high traffic loads and in large-scale deployments.

This study implemented a hybrid routing protocol combining features of proactive and reactive protocols. Found that while the hybrid approach reduced latency and improved packet delivery rates, it still faced issues related to energy consumption and scalability in extensive agricultural fields.

Despite the advancements made by these studies, several gaps remain:

Energy Efficiency: Many existing protocols do not adequately address the energy constraints of sensor nodes, particularly in large-scale deployments where battery replacement is impractical [36].

Scalability: The performance of many proposed routing algorithms degrades as the network size increases, a critical issue for extensive agricultural fields [37].

Adaptability: The ability of routing protocols to adapt to changing environmental conditions and node energy levels remains limited [38–40].

The aim of the research lies in designing modified OSPF routing algorithms that are energy efficient. To achieve the research aim, the following objectives have been specified.

(1) To study the relationship between the network traffic load and power consumption.
(2) To propose a modified OSPF routing technique based on router power consumption.
(3) To analyze and compare the proposed modified OSPF routing technique with existing OSPF routing algorithm.

The contributions of this research include:

(1) Development of a novel modification to the OSPF routing algorithm that prioritizes energy efficiency
(2) Comprehensive analysis of the relationship between network traffic load and power consumption in IoT networks
(3) Demonstrated improvements in energy consumption and network longevity compared to traditional OSPF routing algorithms through simulation results

The remaining of this paper has been organized as follows: Section II is the literature review, Section III is the methodology, Section IV is the experimental set-up and configuration, Section V is the software tools, Section VI is the results and discussions, and Section VII is the conclusion and references.

II. Literature Review

With the rapid advancements in Internet of Things technology, the need for energy-efficient solutions with small packet sizes has become increasingly important. This is especially crucial in IoT applications where devices are often powered by batteries and need to operate for extended

periods without recharging. By optimizing packet size and energy consumption, IoT devices can effectively conserve power and extend their operational lifespans [1–3].

Studies on packet size and its impact on energy efficiency in routers have produced numerous significant discoveries. It has been discovered that energy consumption is mainly influenced by improper configuration and parameter settings, leading to an increase in the size of packet generation [4]. HTTP packets demonstrate efficient power consumption for a limited number of transactions, particularly when dealing with small packet sizes in conjunction with LTE networks. This has been observed in end devices engaged in REST-based resource retrieval using both HTTP and CoAP across various network configurations [5]. Talavera *et al.* [6] focused on routing optimization in power packet dispatching systems, to minimize energy loss. Meanwhile, Rose *et al.* [7] introduced an energy-efficient routing mechanism for wireless sensor networks that aimed to prolong network lifetime and minimize delay by utilizing various techniques and algorithms. Lambrechts and Sinha [8] demonstrated a 35% reduction in power consumption with a "reduced buffer with small packet" scheme in passive optical networks. Puchiano conducted a comparative study of power consumption on two commercial routers, considering two types of traffic: UDP and TCP. The traffic load was generated using the iPerf3 tool. It was found that power consumption increases with the complexity of the protocol [9].

In our experiments, we used two MikroTik routers to measure the traffic going through point-to-point wireless links. The experimental setup involved the use of Mikrotik routers to evaluate the proposed OSPF routing algorithm in a controlled environment. Initially, two Mikrotik routers were used for preliminary testing. This setup allowed for the evaluation of basic routing functionality, energy consumption metrics, and the performance of the modified algorithm under limited conditions. However, to fully capture the essence of routing and to simulate conditions that closely resemble real-world agricultural deployments, a larger number of routers is necessary. Therefore, future experiments will involve more than four routers to create a more complex network topology. This expanded setup will provide a comprehensive evaluation of the routing algorithm's performance, including aspects such as network scalability, load distribution, and energy efficiency across multiple nodes.

iPerf is used to inject the TCP traffic flow to the iPerf server through wireless links. IPv4 is used for all network configurations, and static routing is implemented. Static routing is configured to minimize the transmission of broadcast and extra packets, which could contribute to increased unwanted traffic on the network.

This research hypothesized that a direct or linear relationship exists between the network traffic load and power consumption. Hence, we present an original routing algorithm designed because of hierarchical, cluster based IoT networks. For a large agriculture field, we need a more extensive size network that involves a more significant number of nodes. For extensive network, we have proposed a modified Open Shortest Path First (OSPF) routing algorithm which has been implemented in hardware using actual or real data. We modified the existing OSPF to give preference to the healthier paths based on the criteria of the total energy available on the path, the path length, and the avoidance of critical nodes. The aim of the research lies in designing modified OSPF routing methods that are energy efficient. To achieve the research aim, the relationship between the network traffic load and power consumption must be studied. We analyzed the energy consumption (cost and power of each node) for the implemented algorithm in the network formation.

A. Agriculture-Based Power Reduction Techniques of Sensors

WSNs contain numerous sensor nodes used to measure ecological phenomena in real-time and transmit information through a wireless module back to the master node. Without wire links, sensor nodes are ideal for numerous applications in a strict setting. A sensor node typically has rechargeable batteries of minimal power and a long-term application challenge. These batteries power the sensor nodes with the required current to sustain the operation of each part of the sensor nodes. The cumulative power usage of the sensor node is the quantities of each factor inside the node (e.g., the sensor, the device, and the radio module). The lifetime of a sensor node is, therefore, the period it takes to drain its batteries within a sustainable threshold of activity. Many academics have established power reduction methods to guarantee an unlimited lifetime of sensor nodes. This section discusses a variety of strategies for reducing the strength of sensors for agricultural applications. There will also be a discussion of the goals, wireless standards, observations, benefits, and disadvantages of each methodology.

1) Sleep/wake strategy

In contrast to transmission devices, wireless transceivers use considerable resources. During receipt and propagation, the energy of the wireless nodes would be largely distributed. Send/wake technique for minimizing energy usage of the RF components on WSNs allows the radio of sensor nodes into sleep mode. During sleep mode, there is no data connectivity. Without any pause, the sensor nodes wake up to gather and send data and go back to sleep mode to store electricity. The sleep/week technique is applied in agricultural applications by duty rotation, MAC, and topology regulation. In the area of agriculture, the MAC is used. Service cycle for various agricultural applications is suggested in many studies to decrease energy use. The solar moisture sensor was still sleep-mode and only enabled when data had to be gathered, to reduce the energy usage of the soil sensor nodes. The effective energy WSN for soil moisture control was submitted in the literature to an irrigation device. This platform requires incredibly low power and can conserve a lot of energy in sleep mode where the transceptor (sleep mode) stays inactive for a long time.

2) Radio optimization

Previous studies reveal that much of the power is discharged from data processing units such as microcontrollers and microprocessors in radiographic

components of WSNs. Several scholars used numerous radio optimization schemes or techniques to reduce the power usage of the RF elements of agricultural sensor nodes including transmitting power management, modulation schemes, and cognitive radio. The TPC device modifies sensor nodes to save resources, stimulate interruption avoidance, and create a communications connection. The RF transmitter power of sensor nodes can be used for the agricultural field, which can be changed to decrease their power consumption depending on the gap between the sink node and sensor nodes calculated. Studies have analyzed the application of TPC to reduce sensor node power usage. The mechanism is assisted by many power levels and various receptor sensitivity levels. Results indicate that the power saving of multiple transmitted power modes can be increased by around 10% compared to the conventional model. Cognitive radio is a smart wireless network that enables efficient selection of the wireless communications channel in the spectrum band. Transmission metrics may be corrected accordingly (e.g., transmitted capacity, carrier frequency, and modulation system).

3) Data mitigation

Another resource usage approach to minimize data transmitted from source nodes to target nodes for agriculture WSNs is data mitigation. Data mitigation. Data reduction may be performed by (i) collecting data, (ii) comparing data, (iii) data rate, and (iv) data-guided processes. Data collection will prolong the existence of WSN sensor nodes by reducing redundant data transfer. The sensed data from adjacent nodes may be spatially related to numerous applications of agricultural WSNs like environmental observations. In this situation, the data mix provides an important way and the quantity of redundant data is minimized and distributed, and the energy used by nodes in the network is minimized. The aggregation of data often decreases latency by minimizing data flow and enhancing the period between the sink and sensor nodes; this method will increase bandwidth exploitation after reducing the number of communications. Data fusion may be conducted from sensor nodes to sink nodes along a path. Nodes can retransmit only the limit, the minimum or medium sum of obtained data on a path. The design of an energy-efficient data collection device based on the WSN represents a major challenge, as it must maintain a compromise between energy usage, latency, protection, data integrity, and fault tolerance. The collection of data in agricultural applications, including humidity, wind, soil temperature, and moisture, will help reduce the energy use in environmental sensor nodes by spatially correlating the field and redundant sensor node data, particularly in the nighttime, where the temperature does not change considerably.

III. Methodology

This section presents the overall methodology employed in this work. The design talks about the requirements of the experiment. The network architecture, configuration, and simulation tool used to design the network was described. The software and hardware system requirements which play a major role for design and execution of the results are also detailed.

A. Research Framework

This work aims to design network modified OSPF routing mechanisms that are energy efficient for agricultural IoT application in rural areas with low internet penetration. A modified OSPF routing protocol to improve the routing in IoT networks using the advantages of OSPF solution was proposed and implemented. The modified OSPF routing protocol will be evaluated in comparison with the existing OSPF routing protocol using actual data on GNS3 emulators. The performance metrics are based on the energy, power, load, throughput, and packet rate.

B. Routing Problem

Routing refers to the process of selecting paths in a network along which to send data packets. This process is crucial in ensuring that data travels efficiently from its source to its destination. Since a clear physical link does not necessarily exist between the source and a data packet target, the packet should be transferred from one intermedial routing node to another until it enters the destination. The path to the packet's source is defined as a multi-hop routing. Suppose that it costs *c(RP)* units of network resources (e.g., energy, wireless bandwidth) to send the data packet from a source over a routing path RP to a destination, where c(RP) is a cost function, the multi-hop routing problem is finding a routing path RP such as c(RP) is minimized. The multi-hop routing problem can be illustrated in Fig. 4, which shows a generic and simple routing scenario where the source S wants to send the data packet p to the destination D.

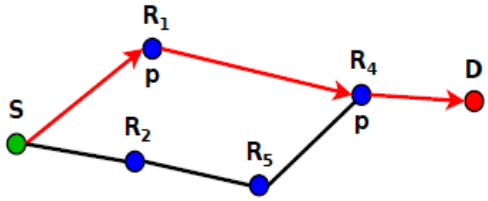


Fig. 4. The general routing problem.

For the sake of explanation, we abstract the network as a graph whose vertices (e.g., S, D, R1) represent routing nodes and edges (e.g., SR1) represent the routing links between them. If we define the cost function as the number of hops, the routing path RPR1; R4 will be selected instead of RPR2; R5; R4 for sending the packet p, because the former requires only two hops as opposed to three hops required by the latter. Specifically, p will be relayed from S to R1, from R1 to R4, and finally from R4 to D.

C. Research Methodology

The objective of most of the routing protocols is to accomplish energy competency using evenly spreading energy among the nodes. In utmost all the present energy competent routing mechanisms, the information is sent away from the point of origin to a target that requires the least energy for transmission. The primary aim of modified routing is to choose the enhanced path by utilizing subsequent metrics.

Average Energy Consumption – This is a correspondence of overall energy utilized by every node to the total incidents sensed.

Load balancing–balance the load equally among the routers.

This methodology will expand the network lifespan.

The proposed modified OSPF routing protocol:

Fig. 5 demonstrates the flow of the modified OSPF routing protocol.

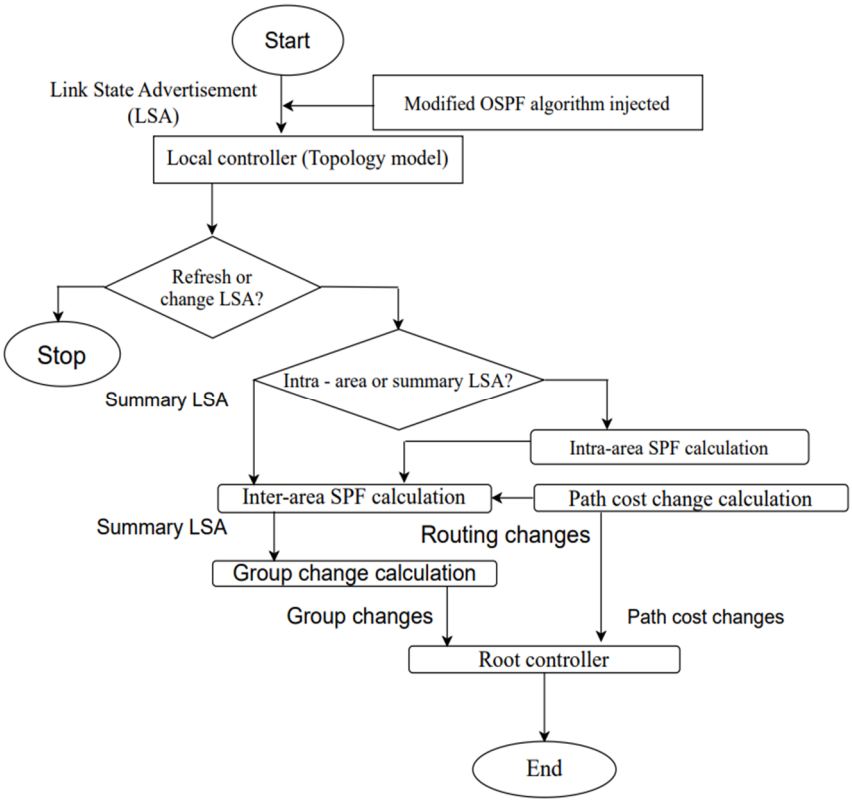


Fig. 5. Flowchart of the modified OSPF routing protocol.

The OSPF Routing Protocol is a cost-based Connection State Protocol instead of hops or ticks (i.e., it is not a vector-based routing protocol). As with RIPv2, it is possible to use various sized subnet masks inside the same network so the usable address space can be more easily utilized. OSPF also promotes point-to-point connections and equal-cost routes (or load balancing for up to 6 paths; meaning balancing the distribution of IP datagrams down parallel routes to the same destination router using a round-robin or a direct addressing option). The key protocol steps can be defined as below:

1) Link State Advertisements (LSA)

Since only link-state ads are shared and not all network knowledge is exchanged (as in RIP), OSPF networks converge much faster than RIP networks. Furthermore, network adjustments cause connection state advertisements (like the triggered updates in RIP). The SPF tree is a CPU-intensive algorithm used in Dijkstra calculations. Therefore, it is advised to operate it in a slot with a slow-speed network or zero.

2) The OSPF process

The Link State Database (LSDB) contains the link state advertisements sent around the 'Area', and each router holds an identical copy of this LSDB. The router then builds an SPF tree using a Dijkstra LSDB algorithm, and you will get a routing table from the SPF tree that now gives the best path to each router. The router is the most common tree. Point-to-point networks and multi-access networks could be possible under OSPF. Each router within a region retains an equivalent LSDB via adjacent contact with other routers. The production of an adjacency takes place between two A and B routers that are in the first downstairs:

Init state: To establish a neighbourly partnership, Hello packets are shared between Routers A and B. They determine whether to become adjacent based on certain packages. The Hello packet comprises the router ID, hi, and dead intervals and is submitted to 224.0.0.5. The hellos are transmitted every 10 seconds in multi-access networks. Normally, the dead interval is four times hi, and the period waiting before the router states that the neighbour is down. The Hello packet requires a 32-bit router ID, usually the highest IP on the router interface or loopback address if configured. When the routers see each other in the Hello packet, bi-directional contact is verified. Often included are the Router Priority, and DR/BDR addresses, and the routers must agree to the flag and authentication password.

3) Two-Way state: The routers link each other to the directory of their neighbours and become neighbours

DR and BDR Election: The router with the highest priority router (information found inside the 'hello' packet) is the DR or router with the highest router ID when an

adjacency is created (highest IP address or the loopback interface address). The BDR becomes the router with the next highest ID. The BDR gets the same data as the DR but does the job of a DR only when the DR struggles. Both routers are always adjacent to the BDR. Both spoken router preferences ought to be set to '0' in a hub and spoken setting so that they can never become the DR or BDR and are thus removed from the other routers. If the network is subsequently added to a higher priority router, it will NOT recognize the DR, nor will re-election occur. In one network, a router will be a DR, and at a time a regular router in another. After the voting, the routers would be in the Ex start state, as the DR and BDR establish a partnership with the routers. Using Database Overview Packets, they start constructing their connection state databases. The exchange method is regarded as an exchange process for the exploration of roads by sharing DBD packets. These bundles include information such as the form of connection, the address of the advertisement router, the cost of the link, and the sequence number, which indicates how fresh the link is. Connection State Advertisement (LCA), which is absent or out of date, is used to evaluate LSDBs Unicasts.

Link State ACK: Once a DBD has been received a Link State ACK is sent containing the link-state entry sequence number. The slave router compares the information, and if it is newer, it sends a request to update.

Link State Request: To update its LSDB, the slave router sends a Link State Request. This is known as the Loading state.

Link State Update: A Link State Update is sent in response to a Link State Request, and it contains the requested LSAs.

Link State ACK: Once a Link State Update has been received a Link State ACK is sent again, and the adjacency has been formed. At this point, the databases are synchronous.

Full: In the Full state the routers can route traffic, and the routers continue sending each other hello packets to maintain the adjacency and the routing information.

D. SPF Calculation

Before the measurement is done, all routers in the network need to be made aware of and linked with all other routers in the same network. The following move is to determine the shortest way for each router. In the Link-State database, they swap link-states with the routers. Whenever a connection state update is obtained on the router, the information is stored in the database, and the modified information is propagated by this router to all routers. The router evaluates the shortest path tree for all the destinations until the database of each router is done. There was a mistake (The shortest path in the SPF algorithm is called the Shortest Path Tree). The shortest path from Dijkstra to the other routers in the networks is then decided by the shortest way from a particular router. The shortest path for each router is then determined at the root of the shortest path tree. The cumulative expense will be the fastest route to hit the destination. The expense of transmitting packets over a certain interface is the metric. The formula to calculate the cost is cost = 100,000,000 /bandwidth in bps. If the bandwidth is wider, the cost would be lower.

E. Assumptions

To formulate the modified OSPF routing protocol for IoT, assumptions considered are explored in this section.

Running Dijkstra identifies the simple station minimal cost direction for the current epoch. The protocol is founded on assumptions:

All equipment begins with the same energy standard.

There is only one base station in the IoT network that is static.

The base station is supposed to provide an endless amount of electricity, i.e., a base station cannot be shut down when the energy is not sufficient (see Table I).

TABLE I. SOME NODE AND NETWORK-LEVEL ASSUMPTION

Node Assumptions	Network Assumptions
All nodes are homogeneous	The entire sensing region partitioned into smaller clusters
Nodes positioning is at random in the sensing area	The cluster head is a router having the high capability, and it is positioned in any area of the target region.
Some nodes are static, and some have limited mobility	For intra-cluster and inter-cluster communication, bidirectional links are used.

F. Implementation of Modified OSPF Routing Protocol

An algorithm had to be built to adjust the interface costs in compliance with resources efficiently. The cost adjustments would automatically trigger an algorithm to run again in Dijkstra, failing a packet. At the same moment, both routers obtain the current network state and upgrade their routing tables. This will just degrade the network output rather than boost it by adjusting OSPF cost too much. In this algorithm, we assume the power consumption will correlate with the traffic. Since this algorithm is to be used for agriculture in rural areas with limited number of routers and other resources, and in this case, we assume we use minimum number of routers. If the traffic is high, the power consumption will be high and vice versa. The method of modified OSPF involves creating a script in RouterOS that runs every minute that will change the OSPF path cost. Once there is a change, it recalculates the SPF algorithm. Fig. 6 displays the flowchart of implementation of the modified OSPF routing protocol.

In one minute, windows, it was determined to check the traffic over the provided connection. When the total traffic is greater than 50% of the full bandwidth of communication during this span, the cost would be increased by 20. If the total traffic is less than 50%, the rate would be decreased by 10. Otherwise, the total traffic will be maintained at 50% bandwidth.

Inside RouterOS itself, it is not feasible to adjust the interface costs dynamically, and a separate software must be built. This software was selected to be built with Java using a Java-based open-source implementation of Gideon LeGrange's Mikrotik API [8]. A configuration file is attached to the application that includes the IP address of each device, and a different IP address for each router

protocol such that only one IP address must be used. The packets are redirected to the correct router with OSPF allowed even if there is no relation of the IP address that fits the selected protocol, given that at least one interface may be reached. The router username and password to make remote connection simpler. The file also describes the CIDR (Classless Inter-Domain Routing), inter-router network addresses, and the corresponding bandwidth of the connection. The initial OSPF expense for a network connection is determined.

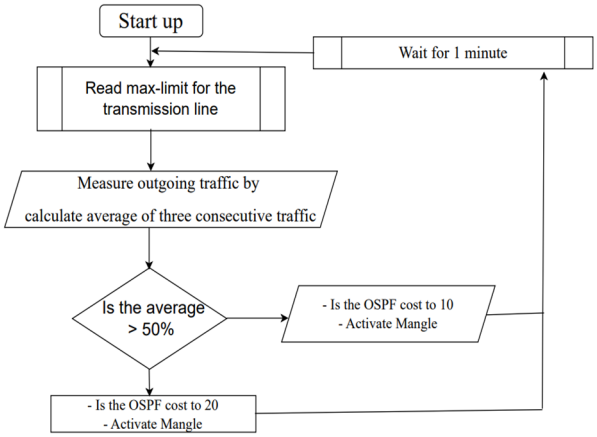


Fig. 6. Modified OSPF routing technique.

After initialization, each router is put in the configuration file and signed on to each router if available. The following move is to retrieve the list of interfaces and IP addresses and to configure a listener for each. To achieve a maximal performance value on that connection, the IP address of each link is often referenced with the network bandwidth values in the configuration file. By contrast, the listener can ask the router in bits per second for the existing bandwidth. As the experiment operates in a virtual environment, bandwidth needs to be controlled as it may affect the performance of the whole experiment. The software tracks the bandwidth every minute, measures the average bandwidth in kilobits per second throughout the time, and uses the above algorithm for the value of the bandwidth contained in the config log. The router on the other side of the network can calculate the same traffic and submit the same result as precisely two routers share each connection. This is constantly replicated, the predicted outcome being that packets are moving around heavily congested areas.

IV. Experimental Set-up and Configurations

A. Building the Topology of the Test Network

The topology of the network is the arrangement of different computer network components. It's normally visually or logically defined. A variety of network component placements, for example, the position of the system and the type of the cabling arrangement, are part of physical topology. The logical topology, on the other side, shows how the data go through the network independent of physical architecture. Network topologies may also be similar, but there are also varying lengths between nodes, physical links, propagation speeds, and signals. The network with each of the three network routers forming a cluster was designed to promote the operation as shown in Fig. 7. This makes ample potential nodes to run numerous traffic scenarios.

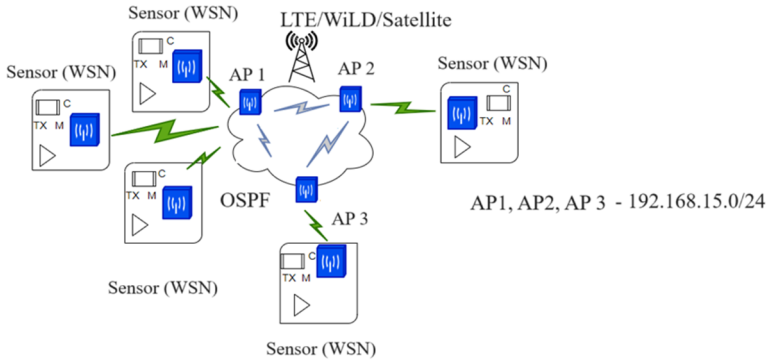


Fig. 7. Map of the test network.

Each router functions from a Mikrotik-base picture, emulated by VirtualBox, RouterOS v6.48.6. Once built, each router has a network interface with /24 addresses for each network connection. On any connection with a default cost of 10 per link, OSPF was configured and enabled. That will ensure that each network connection handles the shortest route representing the least number of hops. There have also been various other 24/7 non-inter router networks, each linked to a parent router, again allowed by OSPF. These networks will make the use of client computers that produce traffic. A bandwidth measurement programme, Iperf, generated traffic. A client device may operate as either a client or a server, with the data produced and transmitted to the server by random data at average speed and registered some packet loss. Either Iperf will saturate the destination with the Total Traffic (TCP mode) or submit a certain amount of traffic (UDP mode).

B. Network and Physical Topology

In the physical topology, the sensor nodes were linked to the routers using a mesh topology. Each node distributes data across the network, and its importance is proportionate to the number of subscribers. The network is fully meshed. This correlation is modelled on the Rule of Reed. Each

node is linked to a completely connected network to establish a complete graph. Therefore, the use of either switching or streaming would not involve a completely linked network. However, in wider networks, because of the quadratically increasing connections with the number of nodes, this network becomes especially impractical. For large networks, then it is extremely unworkable. Technically, a two-node network is completely linked. The real experiment in this project, as seen in Fig. 8.

Fig. 8. Actual set-up of the test network.

C. Field Measurement Setup and Configuration for Relationship between Network Traffic and Power Consumption

The field measurement setup and configuration of the network and the physical topology is illustrated in Fig. 9.

The PTP-Bridge settings and configuration with changes in a blue character, as shown in Table II.

TABLE II. PTP-BRIDGE CONFIGURATION

Wireless Interface	
Interface <WLAN 1>	Wireless
Mode	ap bridge
Band	5GHz-A
Channel Width	20 MHz
Frequency	5805 MHz
SSID	PtP
Frequency Mode	Regulatory domain
Country	Malaysia
Installation	Any
Antenna Gain	0 dB
Wireless Data Rates	
Interface <WLAN 1>	Data Rates
Rate	Configured
Supported Rates B	1 Mbps
Supported Rates A/G	6 Mbps
Basic Rates B	1 Mbps
Basic Rates A/G	6 Mbps
Wireless Transmit Power	
Interface <WLAN 1>	TX power
TX Power Mode	All rates fixed
TX Power	20 dBm
Address List	
Address Network	Interface
10.0.4.2/30/10.0.4.0	Ether 1
10.0.5.1/30/10. 0..5.0	WLAN 1

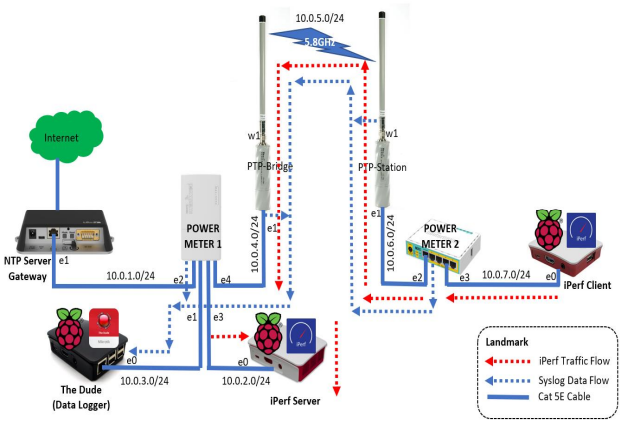


Fig. 9. Field Measurement setup (network and physical topology).

The PTP-Station settings and configuration with changes in a blue character, as shown in Table III.

TABLE III. PTP-STATION CONFIGURATION

Wireless Interface			
Interface <WLAN 1>	Station		
Mode	ap bridge		
Band	5GHz – A		
Channel width	20 MHz		
Frequency	5805 MHz		
SSID	PtP		
Frequency mode	Regulatory – domain		
Country	Malaysia		
Installation	Any		
Antenna Gain	0 Db		
TX power	20 dBm		
Address List			
Address/Network	**Interface**		
10.0.4.2/30/10.0.4.0	Ether 1		
10.0.5.1/30/10.0.5.0	WLAN 1		
Routing table			
Routes	**Destination Address**	**Gateway**	**Distance**
AS	0.0.0.0/0	10.0.5.1 reachable WLAN 1	1
DAC	10.0.5.0/30	WLAN 1 reachable	0
DAC	10.0.6.0/30	Ether 1 reachable	0
AS	10.0.7.0/24	10.0.6.2 reachable ether 1	1

Power Meter 1 and Power Meter 2 settings and configuration with changes in a blue character, as shown in Table IV.

TABLE IV. POWER METER 1 AND 2 CONFIGURATION

IP Address		
Address	**Network**	**Interface**
10.0.1.2/30	10.0.1.0	Ether 2 – gateway
10.0.2.1/30	10.0.2.0	Ether 3-I Perf – Server
10.0.3.1/29	10.0.3.0	Ether 1
10.0.4.1/30	10.0.4.0	Ether 4-PtP
SNTP Client (Enabled)		
Mode	**Unicast**	
Primary NTP Server	10.0.1.1	
Secondary NTP Server	0.0.0.0	
Server DNS Server	–	
Dynamic Server	–	
Poll interval	900s	
Active Server	10.0.1.1	
Last update from	10.0.1.1	

Last Update		00:14:30 ago	
Last Adjustment		38.685 microseconds	
Routing table			
Route list	**Destination Address**	**Gateway**	**Distance**
AS	0.0.0.0/0	10.0.1.1 reachable ether 2 – gateway	1
AS	10.0.0.0/8	10.0.4.2 reachable ether 4 – PtP	1
DAC	10.0.1.0/30	Ether 2 gateway reachable	0
DAC	10.0.2.0/30	Ether 3 – I Perf – Server reachable	0
DAC	10.0.3.0/29	Ether 1 reachable	0
DAC	10.0.4.0/30	Ether 4 PtP reachable	0
IP Address			
Address	**Network**	**Interface**	
10.0.6.2/30	10.0.6.0	Ether 2 – PtP	
10.0.7.1/30	10.0.7.0	Ether 3 – i Perf - Client	
Routing table			
Routes	**Destination Address**	**Gateway**	**Distance**
AS	0.0.0.0/0	10.0.6.1 reachable	1
DAC	10.0.6.0/30	Ether 2 PtP	0
DAC	10.0.7.0/30	Ether 3 – i Perf Client reachable	0

The NTP Server / Gateway settings and configuration with changes in a blue character, as shown in Table V.

TABLE V. NTP SERVER / GATEWAY

IP Address			
	Address	**Network**	**Interface**
	10.0.1.1/24	10.0.1.0	Ether 1
D	10.49.22.210	10.49.22.210	LTE1
D	10.88.200.12	10.88.200.1	L2Tp-Out1
	192.168.88.1/	192.168.88.0	Bridge 1
NTP Server			

	Destination Address	**Gateway**	**Distance**
DAS	0.0.0.0/0	Lte1 reachable	2
AS	10.0.0.0/8	10.0.1.2 reachable ether 1	1
DAC	10.0.1.0/24	Ether 1 reachable	0
AS	10.3.0.0/16	192.168.88.3 reachable bridge 1	1
DAC	10.49.22.210	LTE 1 reachable	0
DAC	10.88.200.1	L2tp-out1 reachable	0
AS	192.168.1.0/24	192.168.88.3 reachable bridge 1	1
DAC	192.168.88.0/…	Bridge 1 reachable	0

D. Measurement Setup and Configuration for Modified OSPF

The GNS3 testbed measurement setup and configuration with network and logical topology are shown in Fig. 10. Once successfully achieved, field measurement will be setup for comparison as depicted in Fig. 11.

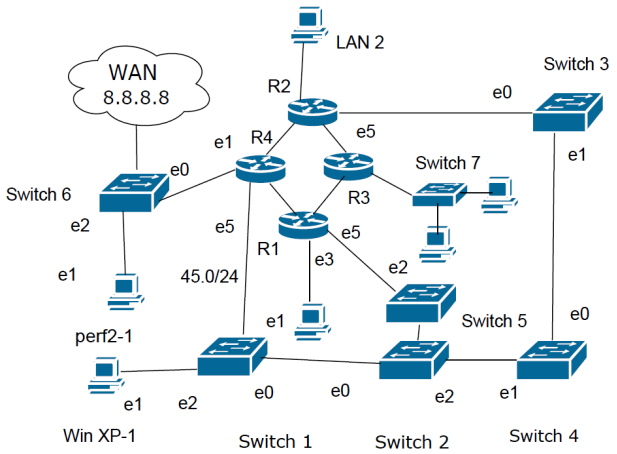


Fig. 10. GNS3 measurement setup (network and logical topology).

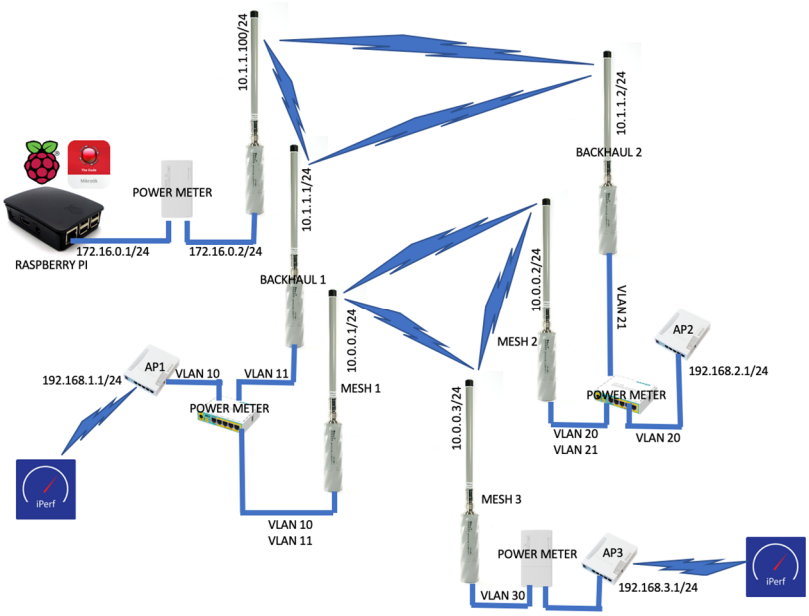


Fig. 11. Field measurement setup for modified OSPF.

For an understanding of real implementation and simulation, Fig. 12 shows the relationship with Fig. 10. Where R1 is AP1, LAN1 is node coming from AP1, R2 is AP2, LAN2 is node coming from AP2, R3 is AP3, LAN3 is node coming from AP3, R4 and WAN is referred to internet source i.e., LTE/WiLD/Satellite. All this device simulates using Mikrotik RouterOS virtual machine. Iperf1-1 and iperf2-1 is an iperf bandwidth test client and bandwidth test server respectively. It's running on DebianOS on top of VirtualBox. WinXP-1 is a GUI

platform for data gathering. It's running on Windows XP on top of VirtualBox.

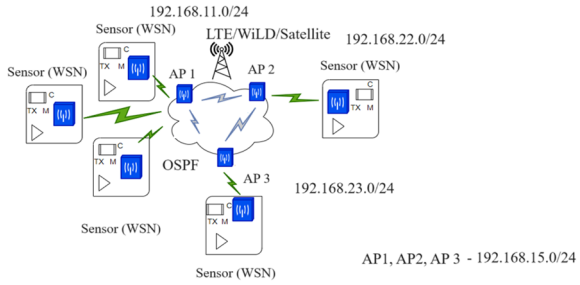


Fig. 12. Real OSPF network implementation.

The router setting and configuration are shown in Figs. 13–16.

CHR-1 [Running]

```
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/ip address
add address=192.168.14.1/24 interface=ether2 network=192.168.14.0
add address=192.168.13.1/24 interface=ether3 network=192.168.13.0
add address=192.168.11.1/24 interface=ether4 network=192.168.11.0
add address=192.168.15.1/24 interface=ether6 network=192.168.15.0
/ip dhcp-client
add interface=ether1
/routing ospf network
add area=backbone network=192.168.14.0/24
add area=backbone network=192.168.13.0/24
add area=backbone network=192.168.11.0/24
/system identity
set name=R1
/tool romon
set enabled=yes
/tool sniffer
set filter-direction=rx filter-interface=ether3 filter-ip-address=\
    192.168.46.6/32 filter-operator-between-entries=and streaming-enabled=yes \
    streaming-server=192.168.15.15
[admin@R1] >
[admin@R1] >
```

Fig. 13. Router 1 and LAN1 settings.

CHR-2 [Running]

```
ing-server=192.168.25.25
[admin@R2] > ip add export
# nov/19/2023 03:13:05 by RouterOS 6.48.6
# software id =
#
#
#
/ip address
add address=192.168.24.2/24 interface=ether2 network=192.168.24.0
add address=192.168.23.2/24 interface=ether3 network=192.168.23.0
add address=192.168.22.2/24 interface=ether4 network=192.168.22.0
add address=192.168.25.2/24 interface=ether6 network=192.168.25.0
[admin@R2] > routing ospf export
# nov/19/2023 03:13:22 by RouterOS 6.48.6
# software id =
#
#
#
/routing ospf interface
add interface=ether2 network-type=broadcast
/routing ospf network
add area=backbone network=192.168.22.0/24
add area=backbone network=192.168.24.0/24
add area=backbone network=192.168.23.0/24
[admin@R2] >
```

Fig. 14. R2 and LAN2 settings.

CHR-3 [Running]

```
    packet-mark=nth-2 passthrough=no
/ip firewall nat
add action=masquerade chain=srcnat disabled=yes out-interface=ether2
add action=masquerade chain=srcnat disabled=yes out-interface=ether3
/ip route
add distance=1 gateway=192.168.13.1 routing-mark=route-1
add distance=1 gateway=192.168.23.2 routing-mark=route-2
/routing ospf interface
add interface=ether2 network-type=broadcast
add cost=20 interface=ether3 network-type=broadcast
/routing ospf network
add area=backbone network=192.168.13.0/24
add area=backbone network=192.168.23.0/24
add area=backbone network=192.168.33.0/24
/system clock
set time-zone-name=Asia/Kuala_Lumpur
/system identity
set name=R3
/tool romon
set enabled=yes
/tool sniffer
set filter-direction=rx filter-interface=ether4 filter-ip-address=\
    192.168.46.6/32 filter-operator-between-entries=and streaming-enabled=yes \
    streaming-server=192.168.35.35
[admin@R3] > _
```

Fig. 15. R3 and LAN3 settings.

CHR-4 [Running]

```
/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/routing ospf instance
set [ find default=yes ] distribute-default=if-installed-as-type-1
/ip address
add address=192.168.45.4/24 interface=ether6 network=192.168.45.0
add address=192.168.46.4/24 interface=ether2 network=192.168.46.0
add address=192.168.14.4/24 interface=ether3 network=192.168.14.0
add address=192.168.24.4/24 interface=ether4 network=192.168.24.0
/ip dhcp-client
add interface=ether1
/ip route
add distance=1 gateway=192.168.46.6
/routing ospf network
add area=backbone network=192.168.24.0/24
add area=backbone network=192.168.14.0/24
/system identity
set name=R4
/tool romon
set enabled=yes
/tool sniffer
set filter-direction=rx filter-interface=ether3,ether4 filter-ip-address=\
    192.168.46.6/32 streaming-enabled=yes streaming-server=192.168.45.45
[admin@R4] > _
```

CHR-10 [Running]

```
  1 8.8.8.8                                       56 254 54ms
  2 8.8.8.8                                       56 254 49ms
  sent=3 received=2 packet-loss=33% min-rtt=49ms avg-rtt=51ms max-rtt=54ms

[admin@WAN] > export
 nov/12/2023 05:17:32 by RouterOS 6.48.6
 software id =


/interface wireless security-profiles
set [ find default=yes ] supplicant-identity=MikroTik
/ip address
add address=192.168.46.6/24 interface=ether3 network=192.168.46.0
/ip dhcp-client
add disabled=no interface=ether1
/ip firewall nat
add action=masquerade chain=srcnat out-interface=ether1
/ip route
add distance=1 dst-address=192.168.0.0/16 gateway=192.168.46.4
/system identity
set name=WAN
/tool romon
set enabled=yes
[admin@WAN] > _
```

Fig. 16. R4 and WAN settings.

The windows XP setting and configuration is shown in Fig. 17.

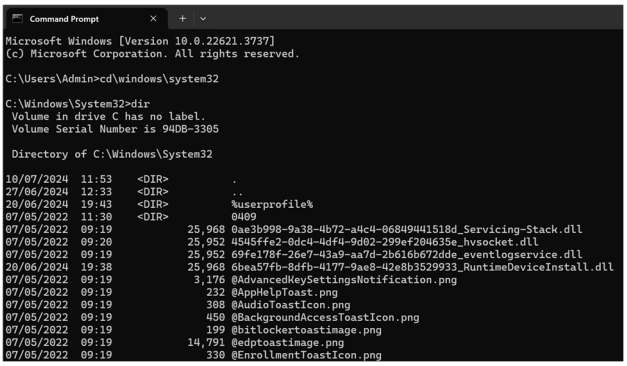


Fig. 17. Windows XP settings and configuration.

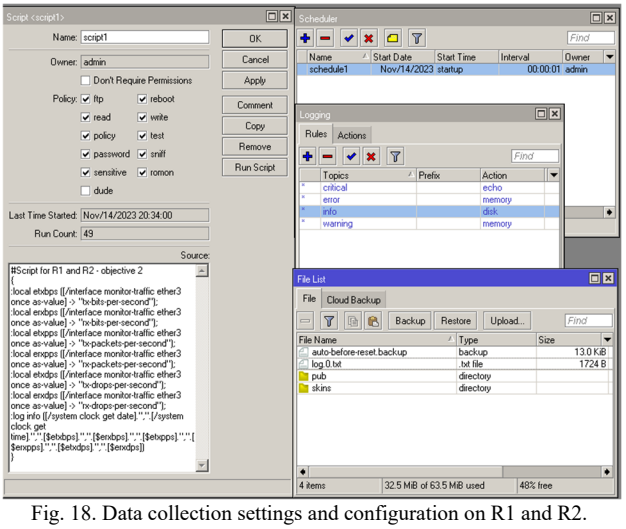


Fig. 18. Data collection settings and configuration on R1 and R2.

Data collection setting and configuration on R1 and R2 is shown in Fig. 18.

As shown in Fig. 15 configuration, the OSPF bandwidth (cost) of R3 has been manually increased by 10. This needs to be done to ensure that traffic destined for the WAN always flows through R2. Means traffic from LAN3 to WAN will always flow through R3→R1→R4→WAN as its total OSPF cost (bandwidth) is 21 (10+10+1), less than R3→R2→R4→WAN which is 31 (20+10+1). Fig. 10 shows the OSPF router of R1, R2, R3, and R4.

V. Software Tools

A. GNS3 Test Bed

To develop a scalable research environment for service, it was agreed not to build a physical network, but to use a network simulator. The GNS3 graphical network emulator package is included. This helps applications to mimic the comportment of a physical network. A graphical user interface is included and equipment such as routers, switches, and links, as in the real network, may be added and can run. GNS3 embraces the most integrated, simple network equipment (e.g., switch) and can simulate hardware by design. Emulation support is implemented and handled via QEMU (Quick EMUlator) or VirtualBox which virtualizes hardware on a single computer once the correct base image file is given.

B. Mikrotik Router

Mikrotik routers as shown in Figs. 19 and 20 run on a piece of software called RouterOS. This is a console GUI that uses text commands to set and modify different functions and parameters on the router.

Fig. 19. Mikrotik router.

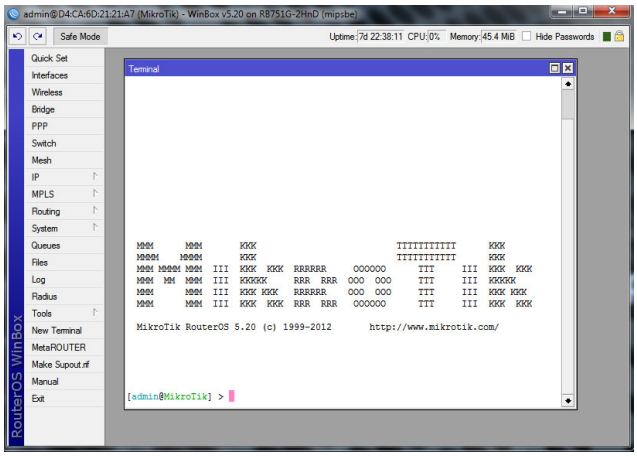


Fig. 20. Mikrotik router interface.

Mikrotik RouterOS router applications for operating system can convert an ordinary personal computer to a specialized router. RouterOS has its API and scripting to allow third-party software and utilities to reach the router.

RouterOS version Cloud Hosted Router (CHR) is designed to run in a virtual computer. Featuring the 64-bit x86 architecture, it may be utilized on the majority of widely used hypervisors, including VirtualBox, Hyper-V, VMware, and KVM.

C. Iperf Tool

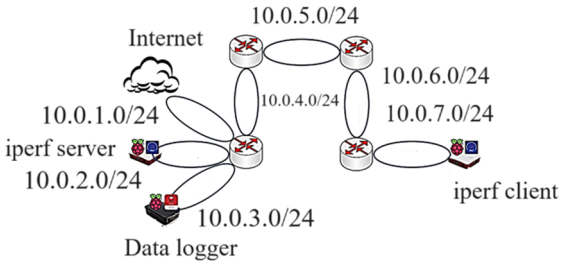


Fig. 21. Raspberry Pi 3 for iPerf server and client.

```
pi@iPerf-Client: ~ $ iperf3 -h
Usage: Iperf [-s] -c host] [options]
Iperf [-h] – help] [-v] –version]
Server or Client:
-p, --port # Server port to listen on/connect to
-f, --format [kmgKMG] format to report: Kbits, Mbits, Kbytes, Mbytes
-i, --interval # Second between periodic bandwidth reports
Full source can be found at:
Iperf3 homepage at: http://software.es.net/iperf/
Report bugs to: https://github.com/esnet/iperf
```

Fig. 22. iPerf command interface.

In Iperf, you can measure the bandwidth in a command-line tool any way you want. You must supply both server and client, compared to online speed checking. In other terms, you link to the database server operated by the test provider while you perform an online speed test (like Ookla). Iperf is a software for comparing network output between two computers. Network output is the sum of usable bandwidth during the measurement. To do this, Iperf recognizes the client and the server as one computer. The client then begins the test with TCP or UDP data streams (the client sends data to the server and calculates how long it takes to send the data; it can then work out the throughput that has been achieved). Figs. 21 and 22 show the Raspberry Pi 3 for iPerf Server and Client.

D. Virtual Box

In our experiment, we create nine virtual router (RouterOS), two DebianOS for iPerf and one Windows XP for GUI and data collection. VirtualBox is a software application that lets we use one physical computer to run same or different operating system. It generates Virtual Machines (VMs) with their own virtual hardware that function like a standalone computer. In our experiment, we create nine virtual router (RouterOS), two DebianOS for iPerf and one Windows XP for GUI and data collection.

However, VirtualBox's smooth operation is limited by your computer's CPU and RAM. To resolve this problem, the measurement speed must be controlled to prevent the CPU and RAM from filling up to maximum capacity. Fig. 23 shows the Virtual Box Manager Window.

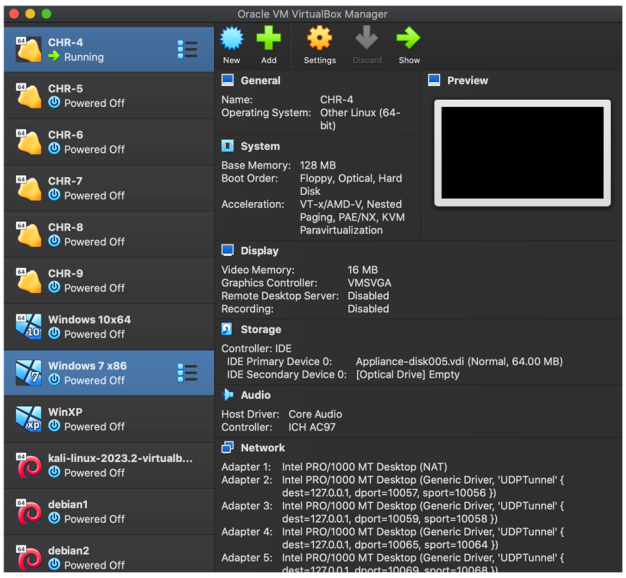


Fig. 23. VirtualBox manager window.

VI. Results and Discussions

The IoT should be intelligent, powerful, efficient, and reliable, with uninterrupted network availability, dynamic QoS management, end-to-end network connectivity to meet the requirements of the characteristics of diversity and dynamics. The evaluation and analysis of the relationship between the network traffic load and power consumption have been presented in this section.

The experimental set-up process used in this research to achieve the entire research is shown in Fig. 24.

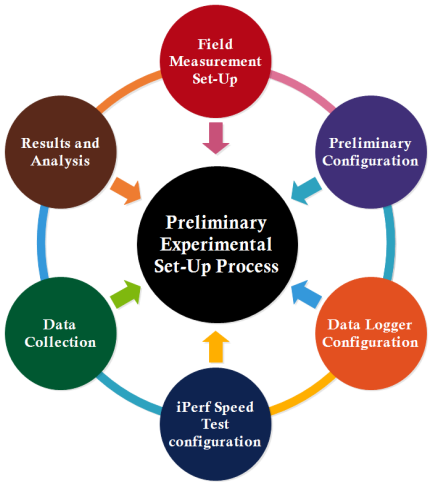


Fig. 24. Experimental set-up process.

The design of IoT for agriculture purposes built on the traditional network architecture has its limitations regarding energy consumption and routing. To address this issue, among other numerous requirements, we present the results from the evaluation and analysis of the relationship between the network traffic load and power consumption. Additionally, we provide results on the modified OSPF once it is applied to the simulation.

A. Data Collection Process

The data collection process used in this research to achieve the entire research is shown in Fig. 25.

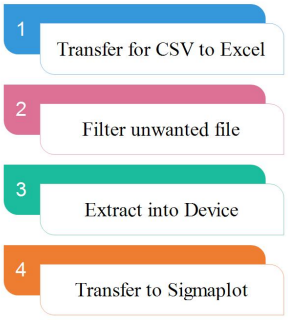


Fig. 25. Data collection process.

It is worth mentioning that before collecting data, we must validate that the wireless medium can support more bandwidth than real traffic. Hence, we ran the iPerf test, as explained in the next section.

iPerf Speed Test Configuration Results, to test our hypothesis, we adapted a relay speed test experiment. Typically, people would like to use "Speedtest" to calculate router speed while confronting the issue of a computer that has a higher speed or speed than the ISP's bandwidth behind the router. However, the outcome of the "Speedtest" may be influenced by the ISP and the network latency, hence Iperf is suggested for research. Iperf is a popular network injection tool that enables TCP and UDP data streams to be produced and a network output evaluated. Iperf helps users to set many criteria for the testing or optimization of a network. Iperf has a client and server functionality and can unidirectionally or bidirectionally calculate throughput between the two ends. Iperf is a server application for clients. The settings and results of the iPerf server and client setup done in this work are shown in Fig 26. This is done before we proceed to the actual experiment to make sure the packet size is correct.

```
[pi@Perf-Server: ~$iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
[pi@iPerf-Client: ~ iperf3 -c 10.0.2.2 -b 1M  -M 960 -t 20
Connecting to host 10.0.2.2, port 5201
[ 4] local 10.0.7.2 port 55638 connected to 10.0.2.2 port 5201
[ID] Interval   Transfer   Bandwidth   Retr Cwnd
[4]    0.00-1.00 136Kbytes 1.11 Mbits/sec 0 23.1 KBytes
```

Fig. 26. iPerf speed test configuration.

The Iperf method used in Fig. 26 was used in calculating full TCP bandwidth, enabling different parameters and UDP properties to be tuned. Bandwidth, delay jitter, and losses from the datagram, results in Fig. 26. The results are TCP measurement and record MSS/MTU scale and read

size found. The results are TCP measurement. Multiple simultaneous links are open to client and server. In Iperf, there's also a UDP where clients may build a certain bandwidth UDP stream and calculate the loss of packets. It tracks delay, jitter and is capable of multicasting, on the other side. If threads are still open, it is multi-threaded. Many links are available for the client and the server. Options K (kilo-) and M (mega-) can be defined where appropriate. So rather than 131072, it is 128K. Instead of a certain number of transfer-data, Iperf may even operate for a certain duration. It chooses the right units for the reporting data scale. The server manages several ties instead of leaving after a single examination. The following must be printed at defined times, intermittent, intermediate bandwidth, jitters, and failure. Iperf is operating the server as a daemon and running it as a Windows NT Service. Finally, it uses representative streams to measure how compression of the connection layer influences your reached bandwidth. Figs. 27 and 28 show the effects of the iPerf speed test before and after position change.

```
pi@iPerf-Client:~ $ iperf3 -c 10.0.2.2
Connecting to host 10.0.2.2, port 5201
[  4] local 10.0.7.2 port 48718 connected to 10.0.2.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   621 KBytes  5.08 Mbits/sec    0   65.0 KBytes
[  4]   1.00-2.00   sec   609 KBytes  4.99 Mbits/sec    0   91.9 KBytes
[  4]   2.00-3.00   sec   512 KBytes  4.19 Mbits/sec    0    117 KBytes
[  4]   3.00-4.00   sec  1.95 MBytes  16.4 Mbits/sec    0    209 KBytes
[  4]   4.00-5.00   sec  2.29 MBytes  19.2 Mbits/sec    0    313 KBytes
[  4]   5.00-6.00   sec   967 KBytes  7.94 Mbits/sec    0    366 KBytes
[  4]   6.00-7.00   sec   510 KBytes  4.18 Mbits/sec    0    380 KBytes
[  4]   7.00-8.00   sec  0.00 Bytes  0.00 bits/sec    0    389 KBytes
[  4]   8.00-9.00   sec   533 KBytes  4.38 Mbits/sec    0    409 KBytes
[  4]   9.00-10.00  sec   288 KBytes  2.36 Mbits/sec    0    492 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  8.19 MBytes  6.87 Mbits/sec    0             sender
[  4]   0.00-10.00  sec  8.07 MBytes  6.77 Mbits/sec                  receiver
```

Fig. 27. iPerf speed test before adjustment of the location.

```
pi@iPerf-Client:~ $ iperf3 -c 10.0.2.2
Connecting to host 10.0.2.2, port 5201
[  4] local 10.0.7.2 port 55510 connected to 10.0.2.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  2.46 MBytes  20.6 Mbits/sec    0    144 KBytes
[  4]   1.00-2.00   sec  2.55 MBytes  21.4 Mbits/sec    0    259 KBytes
[  4]   2.00-3.00   sec  2.37 MBytes  19.9 Mbits/sec    0    375 KBytes
[  4]   3.00-4.00   sec  2.30 MBytes  19.3 Mbits/sec    0    491 KBytes
[  4]   4.00-5.00   sec  2.21 MBytes  18.6 Mbits/sec   10    376 KBytes
[  4]   5.00-6.00   sec  2.22 MBytes  18.6 Mbits/sec    0    424 KBytes
[  4]   6.00-7.00   sec  2.21 MBytes  18.5 Mbits/sec    0    457 KBytes
[  4]   7.00-8.00   sec  1.91 MBytes  16.0 Mbits/sec    1    331 KBytes
[  4]   8.00-9.00   sec  2.66 MBytes  22.3 Mbits/sec    0    359 KBytes
[  4]   9.00-10.00  sec  2.23 MBytes  18.7 Mbits/sec    0    383 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  23.1 MBytes  19.4 Mbits/sec   11             sender
[  4]   0.00-10.00  sec  22.7 MBytes  19.1 Mbits/sec                  receiver

iperf Done.
pi@iPerf-Client:~ $ iperf3 -c 10.0.2.2 -u
Connecting to host 10.0.2.2, port 5201
[  4] local 10.0.7.2 port 59343 connected to 10.0.2.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec   120 KBytes   983 Kbits/sec  15
[  4]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   6.00-7.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   7.00-8.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  16
[  4]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  16
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-10.00  sec  1.24 MBytes  1.04 Mbits/sec  1.586 ms  0/159 (0%)
[  4] Sent 159 datagrams
```

Fig. 28. iPerf speed test after adjustment of the location.

B. Results and Discussions for Relationship between Network Traffic and Power Consumption

This section discusses the results of the simulations done for looking the relationship between network traffic and power consumption. A summary of the table of packet injection rate parameters is presented Table VI.

TABLE VI. PACKET INJECTION RATE (TCP)

Packet Size	Minimum	Maximum	Duration
200 bytes	1M	20M	5 s
500 bytes	1M	20M	5 s
1000 bytes	1M	20M	5 s
1500 bytes	1M	20M	5 s

C. Scenario A: Results with Fixed Packet Size

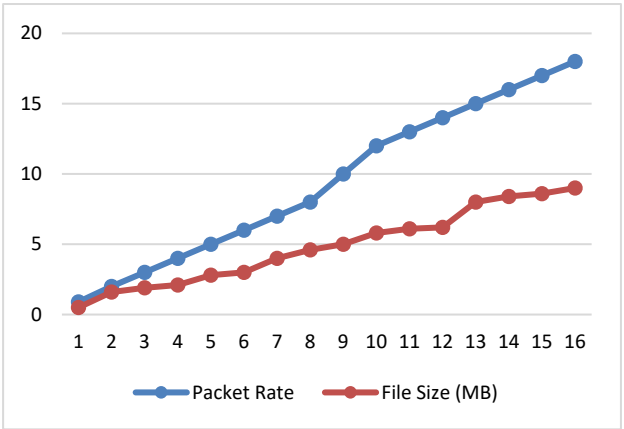
Here we have a situation where 200, 500, 1000, and 1500 worth of traffic flowing in the network is fixed for each simulation instance. A summary of the table of simulation parameters is presented in Table VII. The results for the relationship between file size, throughput, power, and packet rate for all fixed packet size considered have been presented in Figs. 4–9 respectively.

TABLE VII. SUMMARY OF SIMULATION PARAMETERS

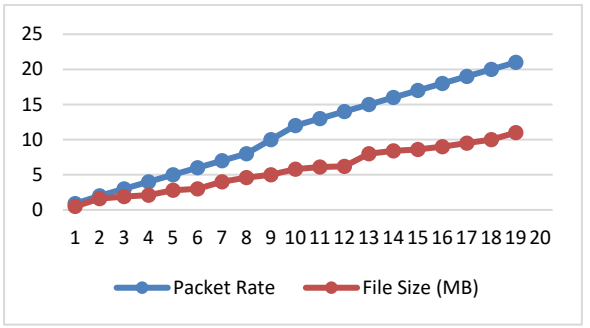
Parameter	Description
Routing Protocol	Static routing
Platforms	Mikrotik RouterOS
Duration	1 hours
Number of Active Routers	2 Routers
Performance Metrics	File Size, Throughput, Power, Packet rate
Simulation Instance	200, 500, 1000 and 1500 Bytes of Data

We first present the results for the relationship between file size and packet rate for 200 bytes packet size. An important characteristic of the IoT network is that despite the small sizes of files being transferred the aggregation of the numerous sensors distributed over a geographical location gives an issue of concern.

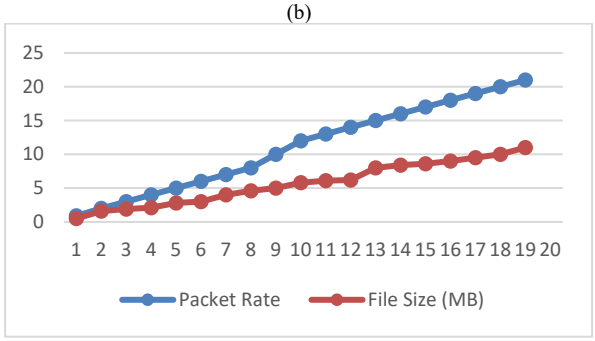
Fig. 29 shows an increase in saturation point as the simulation instance increases from the first to the fourth. From Fig. 29, the network shows saturation at 5 Mbps point, 10 Mbps point, 15 Mbps point, and 20 Mbps point for the fixed packet size of 200, 500, 1000, and 1500 bytes respectively. It can be seen from the results that the throughput of TCP traffic has a slow upward trend with the packet increasing, but this phenomenon deviated with our intuitive understanding. The larger each packet, the smaller that fixed overhead becomes a percentage of the whole. Therefore, larger packets increase throughput. The results for the relationship between throughput and packet rate for 500 bytes packet size is shown in Fig. 30.



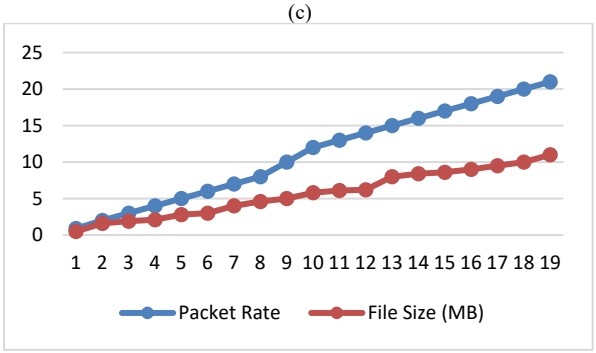
(a)



(b)

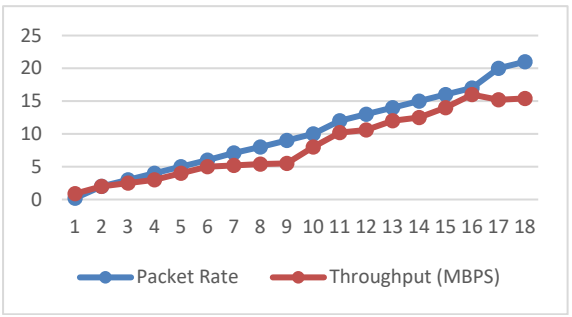


(c)

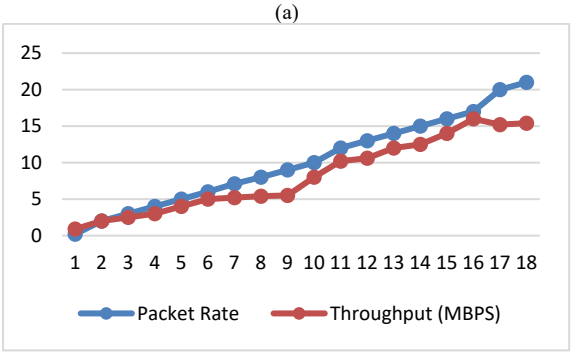


(d)

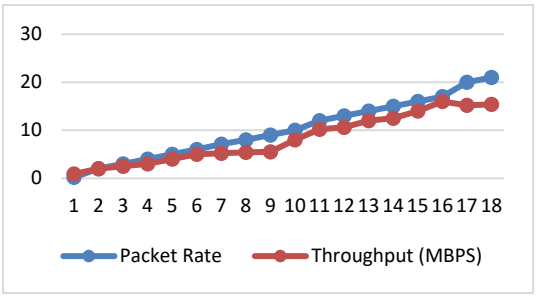
Fig. 29. Relationship between file size and packet rate for all fixed packet size considered: (a) simulation instance 1–200 bytes, (b) simulation instance 2–500 bytes, (c) simulation instance 3–1000 bytes, (d) simulation instance 4–1500 bytes.



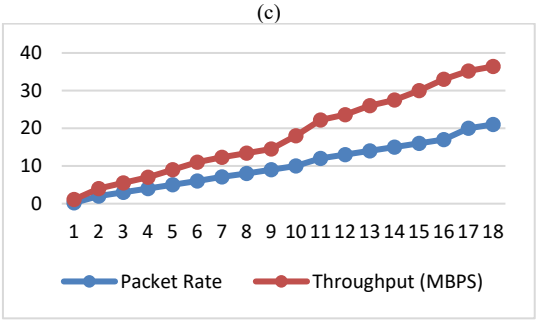
(a)



(b)



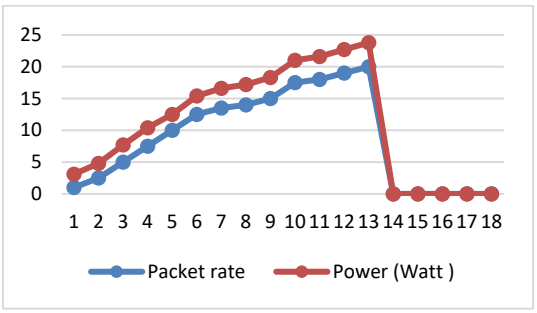
(c)



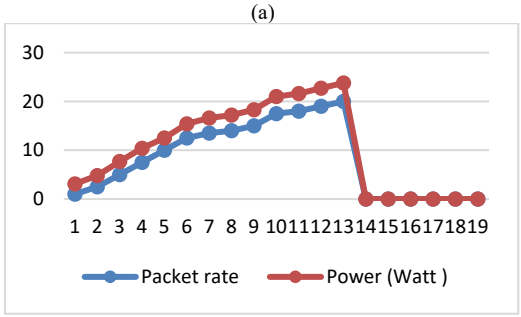
(d)

Fig. 30. Relationship between throughput and packet rate for all fixed packet size considered: (a) simulation instance 1–200 bytes, (b) simulation instance 2–500 bytes, (c) simulation instance 3–1000 bytes, (d) simulation instance 4–1500 bytes.

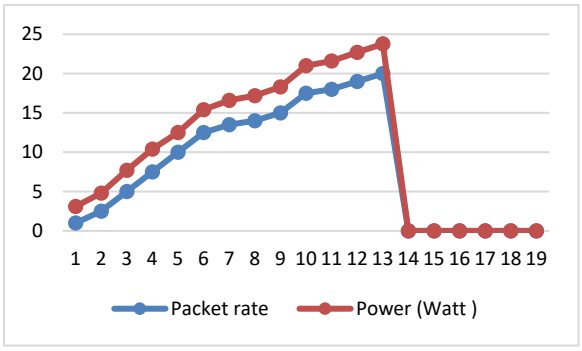
With the rise of the packet failure rate, technically, the throughput declines. Furthermore, the first theory demonstrates the correlation between the packet length and the packet loss rate and changes the pattern between the packet length and packet loss rate. The implementation of this is that each packet has some fixed overhead (e.g., the source and destination addresses). The performance relates to the average success rate of transmission, taking into consideration factors such as overhead transmission, the inefficiency of the protocol and likely rivalry. We calculated how many packets arrive successfully in a set packet size state to their destinations. The capability of the transmission is calculated often in bits per second but can also be measured in data per second.



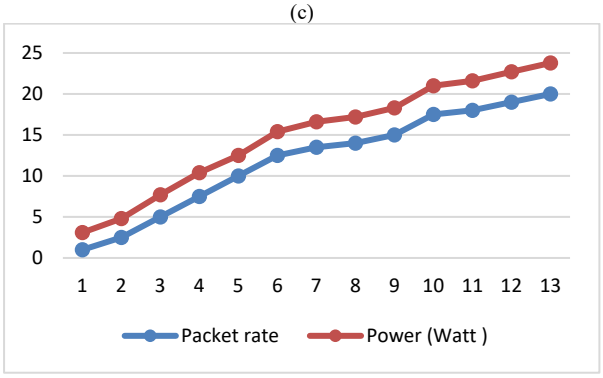
(a)



(b)



(c)



(d)

Fig. 31. Relationship between power and packet rate for all fixed packet size considered: (a) simulation instance 1–200 bytes, (b) simulation instance 4–500 bytes (c) simulation instance 3–1000 bytes, (d) simulation instance 4–1500 bytes.

In discussions concerning TCP, the word "good output" often refers to the sum of accessible data delivered to the receiving application, which may also be regarded as "application-layer passing". Specifically, the data retransmitted may only be counted once for successful results but can be counted twice for certain "performance" meanings. From Fig. 30, the network shows saturation at 3 Mbps point, 10 Mbps point, 15 Mbps point, and 20 Mbps point for the fixed packet size of 200, 500, 1000, and 1500 bytes respectively. Additionally, results for the relationship between file size and packet rate for 500 bytes packet size is shown in Fig. 31.

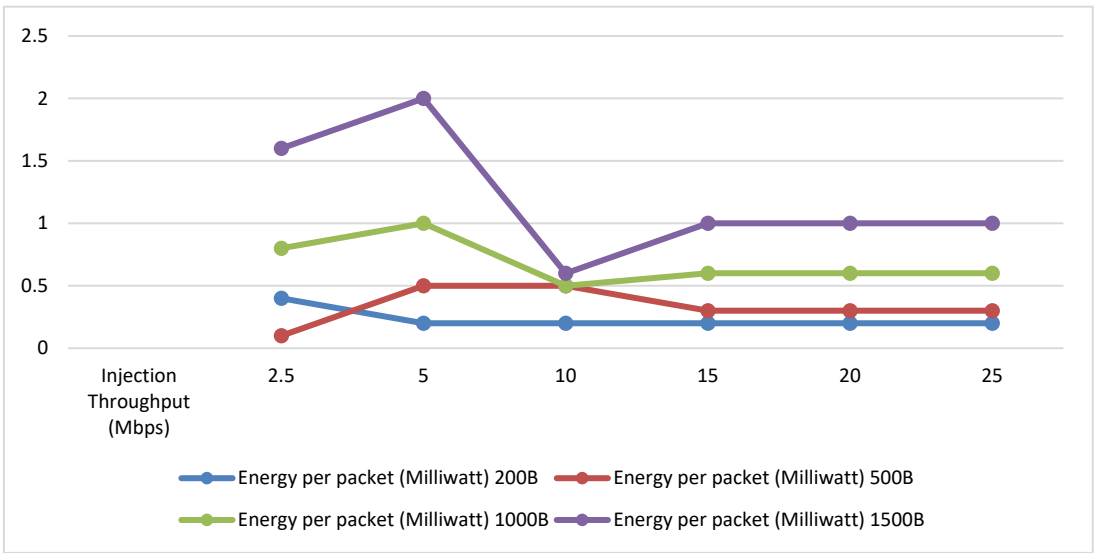
There is an even more critical matter of energy use in wireless networks. The wireless specification specifies several protocols for data transfer, each with different features such as power consumption. It is, therefore, necessary to select the optimum data rate in terms of power consumption and throughput, particularly concerning multi-hop data transmission.

Fig. 32 indicates a different packet size energy usage with different data speeds. It should be remembered that widespread Wi-Fi data transmission energy use consists of two parts: WNI and Processor energy use and storage while copying and processing data. We also remember that calculated power values include the expense of processing the network protocol. Therefore, the relation must be utilized to the utmost to obtain the optimum usage of electricity. A lower MTU is a packet size cap and little more. It has no minimum. A max. If a host sends an intermediate connection over a packet too big, one of the two items will happen.

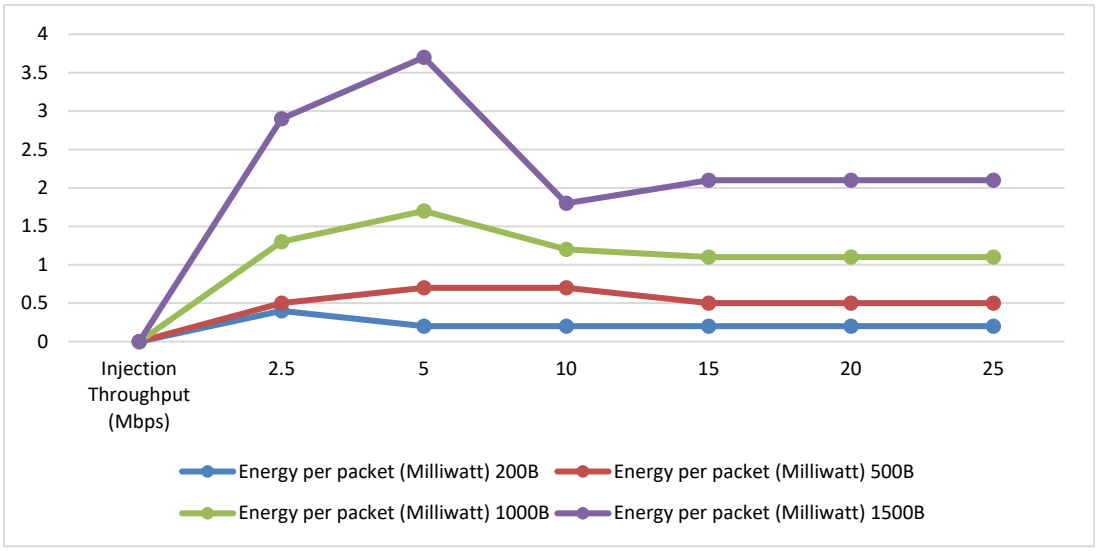
Finally, a big MTU may induce high output on a clean channel, but poor performance can be induced on a noisy channel. Future studies will adopt the principle of lowering MTU sizes to improve efficiency to solve this issue in networks. As a rule, a greater MTU size raises the volume of the protocol. As the MTU size is raised, fewer packets of the same data number can be read. On the reverse hand, a smaller MTU would contribute to additional overhead and approval to be submitted and accepted.

D. Scenario B: Results with Multi Packet Size

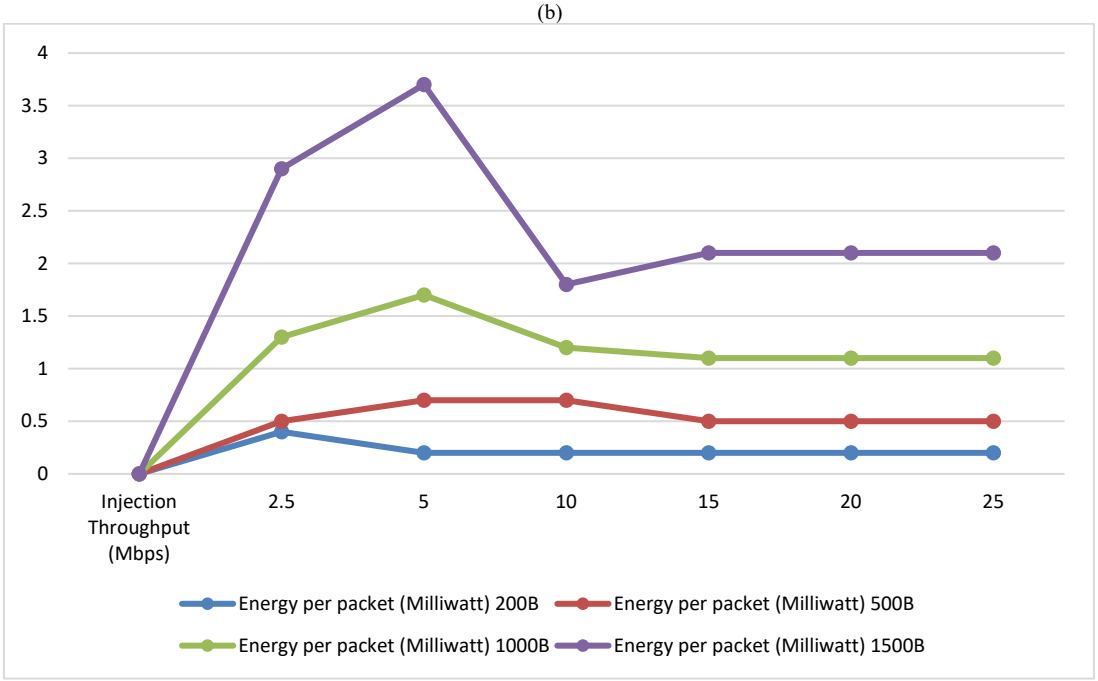
This time we have the same initial starting point as Scenario 1. A summary of the table of simulation parameters is presented in Table VIII. However, this time there is already an amount of traffic flowing in the network. Once the program kicks in and increases the packet size from 200 to 1500, worth of traffic flowing in the network is fixed for each simulation. The results for the relationship between file size, throughput, power, energy per packet, and packet rate for all packet sizes considered have been presented in Fig. 32.



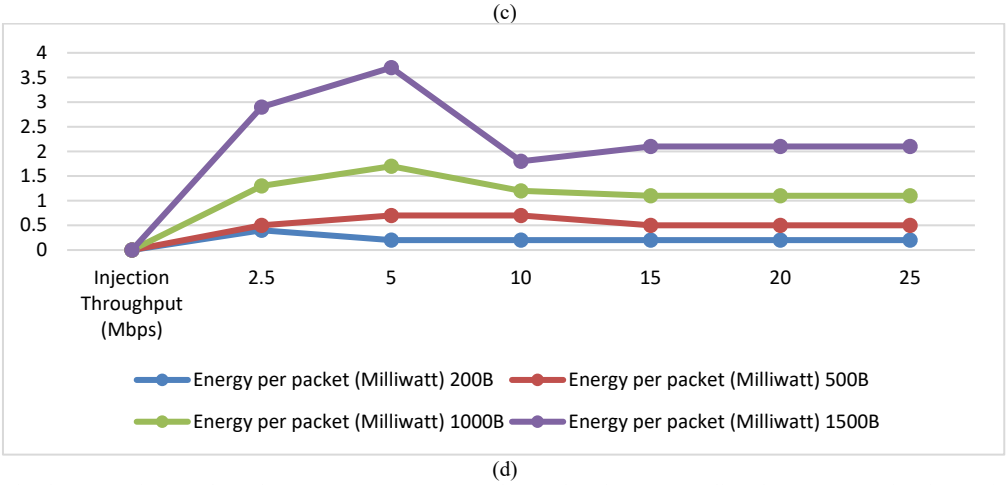
(a)



(b)



(c)



(d)

Fig. 32. Relationship between file size, throughput, power, energy per packet, and packet rate for all packet sizes considered: (a) injection throughput vs transferred size (MB), (b) injection throughput vs actual transmitted throughput (MBPS), (c) power vs injection throughput, (d) energy per packet vs injection throughput.

The comparison of the results from multiple packets shown in Fig. 32 has the same trends as earlier results. Also, from the result in Fig. 32, the saturation point increases as the packet size increases across all metrics considered. The MTU corresponds to but not the maximum frame size, which can be conveyed on the layer of the network connection, for example. Ethernet frame. Ethernet system. Larger MTU has an overhead drop. Fewer MTU values may reduce the delay in networking.

MTU also relies on the underlying network capacity and must be manually or automatically modified to ensure that it does not surpass these capabilities. In combination with a contact interface or norm, MTU parameters can be seen. Some devices will prefer MTU while linking.

E. Experimental Setup for Modified OSPF

The experimental setup for modified OSPF is discussed as follows where GNS3 testbed measurement setup and field measurement setup will be used. Table VIII shows the summary of GNS3 simulation parameters for modified OSPF.

We limit the transmission speed as it is in virtual environment to 100 kbps. Otherwise, the CPU will suddenly increase rapidly as more than 10 virtual devices are running simultaneously.

By default, traffic from LAN3 to WAN will be pass through R3→R1→R4→WAN as its total OSPF cost (bandwidth) is 21, less than R3→R2→R4→WAN which is 31. When the traffic reaches the threshold, the cost of path to R1 will be increased to 10 and it will allow the traffic to do the load balancing.

iPerf will continuously send packet size of 200 byte for a period. The first result is the packet injection with existing OSPF network while the second is using modified OSPF routing technique.

Figs. 33 and 34 show the Iperf speed test with existing OSPF and Iperf speed test with modified OSPF.

TABLE VIII. SUMMARY OF SIMULATION PARAMETERS FOR MODIFIED OSPF

Parameter	Description
Routing Protocol	Existing and modified OSPF routing technique
Platform	MikroTik RouterOS running on GNS3 (virtual machine environment)
Data Duration	Every second for duration of 5 minute
Number of Active Router	5 routers (R1, R2, R3, R4, WAN)
Performance Metric	Packet rate, throughput
Simulation instance	200 Bytes of Data
Limitation	Traffic going to WAN is limit to 100kbps due to virtual cpu resources

Fig. 33. Iperf speed test with existing OSPF.

Fig. 34. Iperf speed test with modified OSPF.

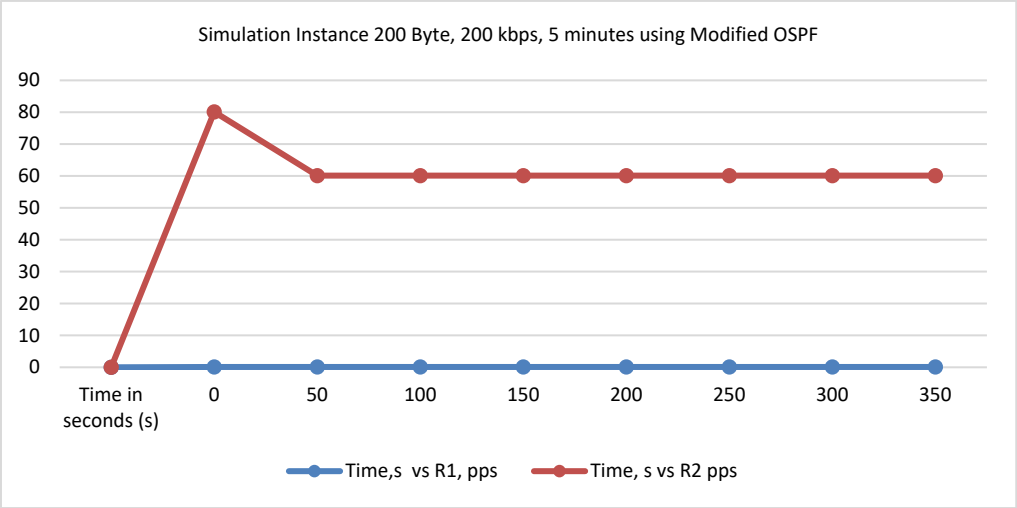


Fig. 35. Received packet rate on R1 and R2 for existing OSPF routing.

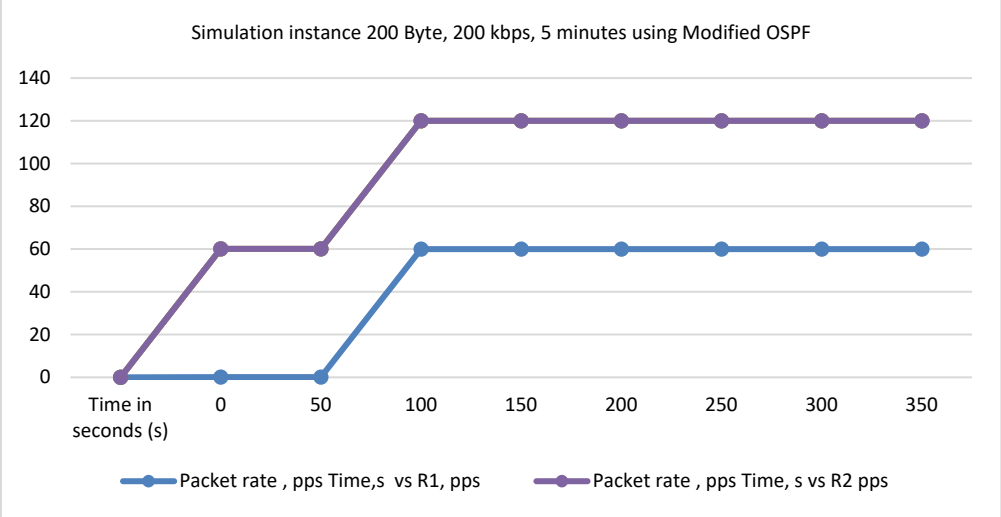


Fig. 36. Received packet rate on R1 and R2 for modified OSPF routing.

F. Results and Discussions for Modified OSPF

This section discusses the results of the simulations done for the first objective of the research. Fig. 35 shows the received packet rate on R1 and R2 for existing OSPF routing technique and Fig. 36 shows the received packet rate on R1 and R2 for modified OSPF routing technique. A performance comparison of the existing OSPF and modified routing protocol is done by evaluating with respect to packet rate, pps vs time. Both the algorithms have been calculated for every one minute. From the figures the performance of modified OSPF routing technique outperforms the existing OSPF routing protocol. The results were verified for a packet size of 200Bytes, data transfer rate of 200kbps with a duration of 5 m.

VII. Conclusion

In this paper, preliminary assessment and review have been given regarding the relationship between network load and electricity use. A successful network architecture can provide the expected applications with an adequate bandwidth. The network does not operate near the saturation stage, because contention is strong, message latency is increased, and overall efficiency is decreased at this period. While the maximum point of the network is reached for a limited time, the output would not decline as seen in the results. Please note that points above saturation are compatible and need a sustained rate of message generation higher than that for saturation. In comparison, the simulation period for these points to achieve a stable state is an order of magnitude higher than for other sections of the plot which indicates that output regression does not happen directly after the point of saturation has been achieved. Even when the message production rate at the saturation point goes down for some time, the output increases again.

A very efficient approach is to restrict the insertion of messages if network traffic is heavy. Traffic can be measured locally for reliability purposes. It is possible to limit the injection by putting a buffer-size limit on the channels of injection, limiting injected messages to use any default virtual channel(s), or waiting until the free performance of virtual channels at the node is greater than the threshold. This framework can be enforced by maintaining an approximation of each router's amount of

free virtual channels. If the amount is greater than the threshold, the injection of the message is permissible. If not, messages must wait at the source queue.

The proposed modified OSPF routing algorithm aims to improve energy efficiency in IoT networks by prioritizing paths with higher available energy, optimal path length, and by avoiding critical nodes, thus extending the network's lifetime.

The complexity of IoT networks, characterized by minimal energy, storage, and varying application-specific requirements, makes designing and implementing an effective routing protocol particularly challenging.

Future work can also be enhanced by focusing on incorporating novel methods of device powering and data exchange, as well as further miniaturization, to advance the emerging concept of data agriculture even further.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Mohammed Khomeini Abu conducted the theoretical and experimental work and was responsible for writing manuscript; Paulson Eberechukwu Numan was responsible for the grammar checking; Kamaludin Mohammed Yusof, Mohamad Kamal Bin A. Rahim, and Mohamad Rijal bin Hamid conducted the literature review; Shaik Mazhar Hussain formatted the manuscript; all authors had approved the final version.

REFERENCES

[1] B. Bajic, A. Rikalovic, N. Suzic, and V. Piuri, "Industry 4.0 implementation challenges and opportunities: A managerial perspective," *IEEE Systems Journal*, vol. 15, 2020.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] G. C. Nelson *et al*., "Food security, farming, and climate change to 2050: scenarios, results, policy options," *Intl. Food Policy Res. Inst.*, p. 1, 2010.

[4] N. H. Bahar *et al*., "Meeting the food security challenge for nine billion people in 2050: What impact on forests?" *Global Environmental Change*, vol. 62, 102056, 2020.

[5] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[6] J. M. Talavera *et al*., "Review of IoT applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, pp. 283–297, 2017.

[7] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *The Internet Society (ISOC)*, vol. 80, pp. 1–50, 2015.

[8] W. Lambrechts and S. Sinha, "Last mile internet access for emerging economies," in *Lecture Notes in Networks and Systems*, Springer, 2019, vol. 77.

[9] P. K. R. Maddikunta *et al.*, "Unmanned aerial vehicles in smart agriculture: Applications, requirements and challenges," arXiv preprint arXiv:2007.12874, 2020.

[10] K. Salemink, D. Strijker, and G. Bosworth, "Rural development in the digital age: A systematic literature review on unequal ICT availability, adoption, and use in rural areas," *Journal of Rural Studies*, vol. 54, pp. 360–371, 2017.

[11] B. E. Whitacre and B. F. Mills, "A need for speed? Rural internet connectivity and the no access/dial-up/high-speed decision," *Applied Economics*, vol. 42, no. 15, pp. 1889–1905, 2010.

[12] M. Mandioma, "Rural internet connectivity: A deployment in dwesa-cwebe, eastern cape, South Africa," University of Fort Hare, 2007.

[13] R. Westerveld, "Inverse telecommunications: The future for rural areas in developing countries?" *Inverse Infrastructures*, vol. 10, 2012.

[14] E. J. Malecki, "Digital development in rural areas: Potentials and pitfalls," *Journal of Rural Studies*, vol. 19, no. 2, pp. 201–214, 2003.

[15] R. Schumann and M. Kende, *Lifting Barriers to Internet Development in Africa: Suggestions for Improving Connectivity*, London: Analysis Mason Limited, 2013, vol. 9.

[16] M. Khalil, Z. Shamsi, A. Shabbir, and A. Samad, "A comparative study of rural networking solutions for global internet access," in *Proc. 2019 International Conference on Information Science and Communication Technology (ICISCT)*, 2019, pp. 1–5.

[17] V. Stocker, G. Smaragdakis, W. Lehr, and S. Bauer, "The growing complexity of content delivery networks: Challenges and implications for the internet ecosystem," *Telecommunications Policy*, vol. 41, no. 10, pp. 1003–1016, 2017.

[18] F. Wu, T. Wu, and M. R. Yuce, "An Internet-of-Things (IoT) network system for connected safety and health monitoring applications," *Sensors*, vol. 19, no. 1, p. 21, 2019.

[19] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014.

[20] S. Mnasri, "Contributions to the optimized deployment of connected sensors on the internet of things collection networks," Université Toulouse le Mirail-Toulouse II, 2018.

[21] S. Rani and S. H. Ahmed, "Multi-hop routing in wireless sensor networks: An overview, taxonomy, and research challenges," *Springer Briefs in Electrical and Computer Engineering*, vol. 17, 2015.

[22] G. Devika, D. Ramesh, and A. G. Karegowda, "A study on energy-efficient wireless sensor network protocols," in *Nature-Inspired Computing Applications in Advanced Communication Networks*, IGI Global, 2020, pp. 158–227.

[23] F. Xia, "QoS challenges and opportunities in wireless sensor/actuator networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, 2008.

[24] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011.

[25] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-Aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[26] M. D. Francesco, G. Anastasi, M. Conti, S. K. Das, and V. Neri, "Reliability and energy-efficiency in IEEE 802.15. 4/ZigBee sensor networks: An adaptive and cross-layer approach," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1508–1524, 2011.

[27] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: A survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

[28] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proc. 3rd ACM International Symposium on Mobile ad Hoc Networking and Computing*, 2002, pp. 146–156.

[29] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad Hoc Networking*, vol. 5, no. 1, pp. 139–172, 2001.

[30] A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[31] M. Malik, M. Dutta, and J. Granjal, "A survey of key bootstrapping protocols based on public key cryptography in the internet of things," *IEEE Access*, vol. 7, pp. 27443–27464, 2019.

[32] F. Pervez, J. Qadir, M. Khalil, T. Yaqoob, U. Ashraf, and S. Younis, "Wireless technologies for emergency response: A comprehensive review and some guidelines," *IEEE Access*, vol. 6, pp. 71814–71838, 2018.

[33] B. U. I. Khan, R. F. Olanrewaju, F. Anwar, R. N. Mir, A. Oussama, and A. Z. B. Jusoh, "Internet of things — The concept, inherent security challenges and recommended solutions," *Smart Network Inspired Paradigm and Approaches in IoT Applications*, pp. 63–86, 2019.

[34] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of things in agriculture, recent advances and future challenges," *Biosystems Engineering*, vol. 164, pp. 31–48, 2017.
[35] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of precision agriculture," *Computers and Electronics in Agriculture*, vol. 157, pp. 218–231, 2019.
[36] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 4, pp. 395–420, 2017.
[37] G. Gardašević, *et al*., "The IoT architectural framework, design issues and application domains," *Wireless Personal Communications*, vol. 92, no. 1, pp. 127–148, 2017.
[38] D. Gao, Q. Sun, B. Hu, and S. Zhang, "A framework for agricultural pest and disease monitoring based on internet-of-things and unmanned aerial vehicles," *Sensors*, vol. 20, no. 5, 1487, 2020.
[39] Q. V. Khanh, N. V. Hau, D. V. Anh *et al.*, "IoT-enabled smart agriculture: Architecture, applications, and challenges," *Applied Sciences*, vol. 12, no. 7, 2022.
[40] V. K. Quy, D. C. Nguyen, D. V. Anh, and N. M. Quy, "Federated learning for green and sustainable 6G IoT applications," *Internet of Things*, vol. 25, 2024.