A Comparative Study on Software-Defined Network with Traditional Networks

Berty Smitha Evelin Zoraida [1], Ganesan Indumathi [2]

[1] *School of Computer Science, Engineering and Applications, Bharathidasan University, India*
[2] *AVVM Sri Pushpam College, Poondi, India*

***Abstract* – The emergence of a technological age, in which nearly everything is linked and accessible from anywhere, is largely due to the Internet. Hence, traditional Networks continue to be difficult and complex to operate despite their widespread deployment. As a consequence, it is challenging to set up the networks in accordance with the established protocols and respond to variations in load and defects by reconfiguring the structure. The data and control layers are packaged together in the contemporary systems, further complicating problems by vertically integrating them. Because of their limited resources and unsecured transmission channel, wireless sensor networks (WSNs) are particularly vulnerable to persistent security attacks. Many hundreds of self-organized and resource-constrained sensor nodes make up the WSN. It is possible to build an ad-hoc network of sensors without a specified architecture or centralised control because these sensor nodes are often organised in a scattered form. The underlying problem is to strengthen the privacy enforcement in these systems because WSNs will have power over real-time applications, where unauthorized conduct might lead to significant harm. In order to address this issue, the software-defined network (SDN) technology was developed. With the creation of SD, network operators now have more legitimacy and influence over the networks.**

Based on a global perspective and centralised management of the network topology, SDN has increased the rigour of its enhancing security. This study describes the software-defined network, outlining its fundamental ideas, how it differs from conventional networking, and its architectural tenets. Also, it highlighted the key benefits of SDN while emphasizing availability, maliciousness, packet delivery and duplication ratio, and efficiency.

***Keywords* – Software-Defined Networks (SDN), Wireless Sensor Networks (WSN),Packet Delivery Ratio, Packet loss Ratio,Malicious Node.**

DOI: 10.18421/TEM131-17
https://doi.org/10.18421/TEM131-17

Corresponding author: Ganesan Indumathi,
AVVM Sri Pushpam College, Poondi, India
Email: ginduavvmspc@gmail.com

Received: 22 September 2023.
Revised: 07 January 2024.
Accepted: 12 January 2024.
Published: 27 February 2024.

1. Introduction

Every aspect of our lives is profoundly impacted by the Internet. A crucial part of TCP/IP connectivity is played by the routing technologies. In order to attain their intended endpoint, the information packets travelling from the origin pass through routers, switches, and other components of the Network. These protocols are responsible for promptly identifying link breakdown and determining an alternate path to the target whenever a link and node disaster happens [1]. In modern environment, the capacity for error identification and remediation has grown crucial due to the rise in unanticipated breakdowns and threats. According to this, it is extremely important to move data coming from an origin to a certain endpoint whenever a link fails or even when modifications to the topological data take place. For the purpose of preventing minimizing packet or data losses in a particular scenario, it is essential that one be able to foresee or know the estimated optimum duration required for a networks to converge [2]. As a crucial efficiency metric and architectural objective for assessing the effectiveness of the routing algorithm, routing convergence rate is thought to be among the most important evaluation criteria [3] and resilience is crucial for networks; the more quickly the protocol's operating gateways assist the system in coming together in the event of a loss, the more dependable it ought to be utilised for real-time applications [4].

A network administrator needs to setup every single access point independently in a limited network design in order to carry out any desired network regulations due to the scattered structure of the system. This would take time and could result in more sophistication or problems being introduced as a result of reconfiguring specific hardware or packet losses. The difficult task for network design is to make the network dynamic and capable of being operated autonomously without the intervention of a physical adjustment. Traditional Internet protocol networks lack the sort of automated hardware and control task modification. Furthermore, in conventional networks, the main communication components are vertically integrated. The phrase "vertically integrated" is employed since a basic networking device comprises both the controlling component, which manages network activity, and the forwarding characteristic, which forwards network traffic in the manner of digital packages as instructed by the controlling attribute. In consequence, this limits the layout options available and forces the creation of a stable framework. In the end, it may serve to limit creative suggestions for the development and utilization of constrained technology networks [5].

The development of Next Generation Networks is being challenged by fresh studies that seek to tackle the constraints posed by existing IP based network design and its supporting technologies [6]. Several strategies are considered for the effective implementation of the Future Network [7]. Numerous research individuals and initiatives have already been conducted by both public and commercial organizations, as stated in [8]. The Software Defined Networking (SDN) framework constitutes one of the most current approaches for the evolving networking concept, among the many contributions made to enhance the rigid traditional architecture network [9]. A programmable system known as Software-Defined Networking offers hope for overcoming the various constraints now faced by conventional network architecture. Furthermore, it eliminates the vertical integration difficulties problem by separating the controlling plane from the actual network core components, such as routers and switches, that forward data traffic. Moreover, with the control plane and the data plane separated, the essential devices are solely in charge of transmitting data traffic at the logically centralised operator's direction. All regulating tasks are now performed by a centralised software-based controllers. The controller in SDN is a type of controlling mechanism that utilizes a sophisticated programmable coding system to regulate the underlying network technologies.

This setting causes the controllers and the software operating inside the controller to be additionally referred to as Network Operating Systems [10].

This article contrasts the efficiency of the standard IP-based network mentioned above with the OpenFlow-based network. Several network nodes are implemented, and the effectiveness of the suggested networks is analyzed by contrasting the outcomes. This study also includes a design for an OpenFlow-enabled campus network design, and evaluations of the proposed network's effectiveness compared to more established networks are done. By connecting Software-Defined Networking nodes to actual infrastructure nodes, a suggested network framework is created. An enhanced open-source simulator such as NS-2 is employed to design the network model. The study is structured with a discussion of a succinct overview of SDN and conventional networks, as well as their structural layout and operation in Part II. Part III compares conventional and SDN networks and analyses the outcomes. Both types of networks are created utilizing an open-source simulation platform that is covered in the same part. In Part IV, it is explained how a simulated node interacts with an actual node, and a performance comparison between a standard network and an OpenFlow-enabled network is conducted according to the presented work. An efficient network framework has been developed utilizing the same open-source technology.

1.1. Background of the Study

Traditional networking employs a distributed approach for the control plane. Address resolution protocol, STP, OSPF, EIGRP, BGP, as well as other protocols function independently for every access point [11]. Although these networking devices are linked together, there is no central machine that oversees or summaries the entire system [12]. The key difference between traditional networking and software-defined networking is that the earlier is focused mostly on hardware, while the latter is typically based on software [13]. Software-Defined Networking is more flexible because to its software foundation, allowing users to more easily manage services when they are remote [14]. Switches, routers, as well as other physical devices are employed in traditional networks to develop relationships and run the networks [15]. In Software-Defined Networking controllers, a northbound interface is utilised for interacting with Application Programming Interfaces [16]. Device makers could actively Programme the networks owing to this connectivity rather than employing the protocols necessary for traditional communication [17].

Traditional networks are employed to physically combine all data layers and control layers, share their resources, boost traffic, and place additional demands on the memory and the central processing unit across 2 procedures [18]. By isolating such operations and having a dedicated server, detachments of control layers and data planes in Software-Defined Networking could be readily tracked and commanded by the controller and networks to perform the required tasks and enabling the networks to effectively setup with a lower collision burden [19]. Since Software-Defined Networking makes it possible for Information technology administrators to offer more physical infrastructural functions and bandwidths without needing to make a financial commitment, it is regarded as a preferred substitute for traditional networking [20]. Traditional networking demands for newer technology to enhance network performance [21]. Thus, a comparative evaluation of the traditional network and SDN was carried out in this research. The outcomes show that the SDN outperforms conventional networks in terms of performance. The conventional network and Software-Defined Networking are illustrated in Figure 2.

1.2. Purpose of this Study

Software-Defined Networking is characterized as a contemporary concept that is quickly replacing conventional networking for systems that cannot eliminate the shortcomings of that architecture by separating software from hardware. With Software-Defined Networking, a centralised software application provides control and management for the devices. The hardware itself is separated from this software package [22]. An open-source structure criterion and layered layout are the primary needs of Software-Defined Networking. Technology seems to be more efficient, more versatile in terms of programming, and more conducive to innovation in information systems since it may be developed by numerous vendors with ease. SDN has a number of concerns that require to be solved, including scalability challenge, virtualization, connection stability, where to put the controllers. One of the major challenges facing SDN is reliability. For expansive networks, stability is a crucial concern. It is theoretically a unified control function in the Software-Defined Networking because the SDN controller is often at one point of failure. Thus, actions must be taken to guarantee that the dependability of contemporary technical alternatives is at least as good as or higher than ever before.

One of the most significant technological advancements for building the network framework of the modern economy is SDN. But the role of technology cannot be built on unstable networks [23]

2. WSN Mechanism

Wireless Sensor Networks (WSNs) have emerged as one of the most cutting-edge approaches to low-energy wireless communications in recent decades. The rapid advancement of low-power wireless technology, the considerable advancement of distributed signal analysis, adhoc networking protocols, and pervasive computing have all worked together to create a coherent perspective for wireless sensor networks [24]. A significant amount of sensory networks are implemented in the most of Wireless sensor networks to collect information according to application fields. This data collecting procedure may be continuous, event-driven, or query-based. Wireless sensor could be employed in a variety of industries and applications, including farming, environment monitoring, tracking wildlife, healthcare, defense surveillance, industrial automation, home automation, and security, among others [25], [26]. Figure 1 depicts a conventional concept of a wireless sensor networks.

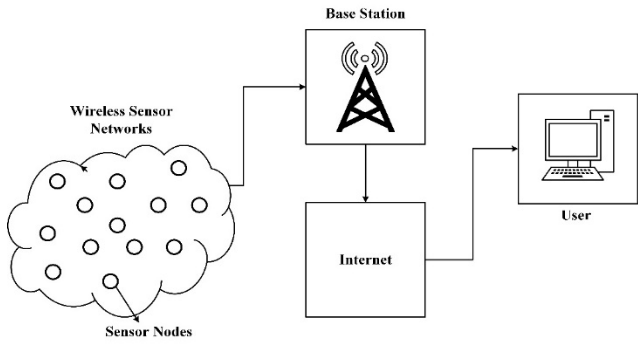

Figure 1. Basic Structure of WSN

For wireless sensor networks, a delay-aware routing technique is offered in this subsection. The protocol attempts to send packets of data to its base station or destination before the anticipated cut-off time. According on the data acquired from its neighboring nodes, the protocol builds and maintains a forwarding database. The following presumptions were considered when developing this procedure [2]. A sensory record contains a wide deployment of homogeneous sensor network. Every sensor node uses various localization algorithms to determine its location. All sensor nodes might receive GPS location updates from the base stations or sink nodes, which are configured. Sensor nodes have very little nodes and are immobile.

All sensor nodes have an equivalent beginning power levels and radio capability. The suggested protocol's principal elements are Neighborhood Management and Packet Forwarding.

2.1. Neighborhood Management

By exchanging HELLO and ACK control packets, every sensor network discovers its one hop neighbours. These are the parameters included in the Hello packets:

- source node id
- source node position
- distance to Sink

Every server transmits an ACK message with the parameters listed below in response to the Hello packet:

- Neighbor node id
- Neighbor node position
- Distance to Sink
- Residual Energy

Figure 3 demonstrates the structure of the HELLO signal and ACK control packets, respectively (Eq 1). Every sensor network keeps a table-like record of this neighbor finding data, which is then modified on a regular basis.

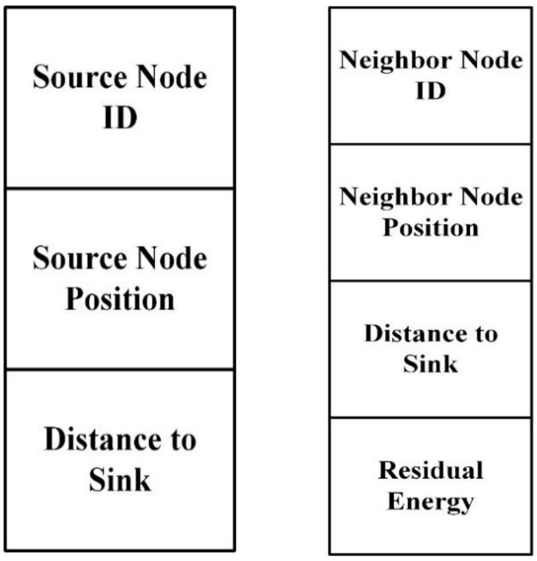

Figure 3.a. Source node parameters

Figure 3.b. ACK packet parameters

Figure 3. Representation of HELLO and ACK packet

Following the completion of the neighbor discovery process, every sensory node determines the link latency to its neighbouring nodes by broadcasting an echo packet and logging the echo packet's round-trip duration.

The link latency is determined for a combination of nodes $(\boldsymbol{r_u}, \boldsymbol{r_v})$according to equation Eq (1)

$$\boldsymbol{Link\ delay_u^v} = \frac{\boldsymbol{Round\ Trip\ time}}{\boldsymbol{2}} = (\boldsymbol{D^{MAC}} + \boldsymbol{D^{Queue}} + \boldsymbol{D^{TRX}}) \times \boldsymbol{A_u^v}$$

Where,$\boldsymbol{D^{MAC}}$: $\boldsymbol{Disruption\ of\ the\ channel}$
$\boldsymbol{D^{Queue}}$: $\boldsymbol{Delay\ in\ queue}$
$\boldsymbol{D^{TRX}}$: $\boldsymbol{Queuing\ and\ transmission\ delays}$
$\boldsymbol{A_u^v}$: $\boldsymbol{Number\ of\ transmissions}$

Depending on the amount of traffic utilizing the network, this latency might change. Every sensor node keeps a packet transmission database that is searched for when determining how to route data packets to their intended location based on the neighbour finding information and projected network latency. The headings of the information sending section are as follows:

- Neighbor node id
- Neighbor node Position
- Distance to sink node
- Link delay
- Energy level

2.2. Packet Forwarding

The sensor network would utilize multi hop communication to forward packets of data to the sink node when they are received as a consequence of an event occurring or when they are obtained from a neighbouring node. In order to do this, every sensor node consults its routing table before sending the data packet to a potential next hop in the direction of the sink node. The operator can set the deadline or the event area's source nodes may determine it. The data from the sensor network is sent as a package with the control parameters shown below:

- source node ID
- sink node ID
- deadline

When this packet arrives, the sensor network chooses the appropriate criteria to determine which node should receive it as its subsequent recipient. The chosen next hop ought to be a little bit nearer to the final location than the present node. The given data packet propagation speed on the chosen link must match the propagation required speed for that network. Depending on the projected schedule, the necessary transmission speed changes. Let $\boldsymbol{r_v}$be a node's neighbour when $\boldsymbol{u} > \boldsymbol{v}$ for node, $\boldsymbol{r_u}$. The distance among nodes r_u and ,r_v is represented by the expression $\boldsymbol{D}(\boldsymbol{r_u}, \boldsymbol{r_v})$. As $\boldsymbol{K^{request}} = \boldsymbol{D}(\boldsymbol{S}, \boldsymbol{T}) / \boldsymbol{t^{set}}$, the source node $\boldsymbol{S}$, determines the necessary end-to-end transmission rate of information for the anticipated timeframe $\boldsymbol{t^{set}}$towards the sink node, $\boldsymbol{T}$.

If the propagation speed, K^{Prop}, on that connection is more than or equivalent to $K^{request}$ and the neighbouring node is nearer to the sink node than that of the present node, the neighbouring node would be chosen as a routing path. The needed and provided transmission speeds in each intermediary node among both the source and sink are determined in the following manner:

t^{set}: The source node's anticipated timeframe for the packets of data.

t^{s}: There is still time to fulfil the deadline. It is equivalent to t^{set}at the source node, where t^{s}.

Each intermediary node updates t^{s}as follows: t^{s}= t^{s}- link latency. The necessary rate is once again determined as per Equation (3)

$$K^{request} = \frac{D(r_u,T)}{(t^{s} - Link\ Delay)} \qquad (3)$$

Here, $r_u$is a node in the way among both the sink and the source that serves as an intermediary. r_u is the source node S when u equal to zero. In a similar manner, the propagation speed supplied on every chosen link is calculated as stated in Equation (4)

$$K^{Prop} = \frac{D(r_u,T) - D(r_{u+1},T)}{(Link\ Delay_u^{u+1})} \qquad (4)$$

If the specified rate, K^{Prop} , is more than or equivalent to the needed speed, $K^{request}$, as well as the forwarding nodes is closer to the endpoint than that of the present network, the forwarding node would be chosen as a forwarding node. If the substitute link satisfies the necessary propagation velocity, a duplicate of the data packets is also transmitted to a different neighbour node. In the event that the primary connection fails, this would be essential for transmitting information packets. Also assisting in assuring dependability is this duplication. Only the source node—that detects the existence of an occurrence—performs this redundant data. Every intermediate node only forwards packets along one path to the final destinations.

3. SDN Mechanism

The SDN Mechanism plays a vital role in networking mainly for 5G and virtualization. When compare to traditional networks ,The SDN supports decoupling in its network architecture with three layers using OpenFlow that can transmit data in communication networks. Their effectiveness is evaluated based on three key metrics: availability, performance, and reliability.

3.1. Architecture of SDN

Software-Defined Networking architecture guarantees the stability and dependability of software and illustrates how SDN functions at various phases.

The three main layers of software-defined networking are as follows: Data plane, Application plane, and Control plane [27]. Software-Defined Networking comprises of two functionalities: one connecting the southbound APIs (such as OpenFlow) while the other is connecting the control plane of the northbound API and the application server of the Application Programming Interface. Figure 3 depicts the two interfaces that make up the Software-Defined Networking [28].

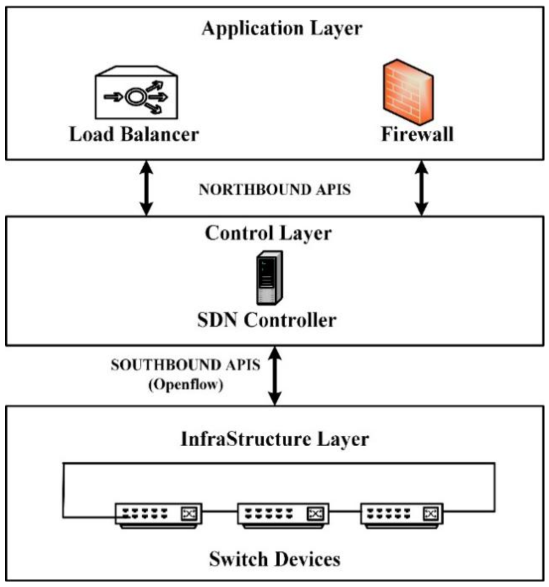

Figure 2. Basic Structure of SDN Framework

3.2. OpenFlow-based on SDN Framework

Software-defined networking varies from conventional networking design in that the control plane and data plane are separated, as was mentioned before. The adoption of OpenFlow technologies at Stanford University led to the creation of software-defined networking. OpenFlow systems were first introduced as a dormitory networks, but in 2011 ONF launched a working collection of software-defined network through open standard implementations by assembling a number of enormous connectivity firms [29]. OpenFlow is being standardised as the initial software-defined networking framework for software-defined design by ONF, which is now supporting both software-defined networking and OpenFlow. As seen in Figure 2, the structure of an OpenFlow-based system consists of 3 levels. The bottom layer, denoted as the data plane, only has the charge of transmitting information packets that arrive at its access control port. The second level, known as the control plane, sits above the data plane and holds the charge of centrally controlling all the underlying packet transmission components utilising centralised control logic.

Application plane, the 3rd and topmost plane, is a potential option. In most cases, a singular controller manages all of the forwarding elements. The application plane's responsibility is to establish the use-cases required to ensure convenient information traffic transfer at the start of network activity. Only tasks relating to handling and information forwarding fall under this layer's purview. On a secure link, utilizing open flow platform as previously stated, the control plane and data plane communicate with one another. Three communication kinds are supported by OFP: asynchronous, controller-to-switch, and symmetrical [30]. Information flow records are stored in forwarding table on the OpenFlow-enabled switch that makes up the information layer. Fig. 3 depicts the elements needed for an OpenFlow system configuration.

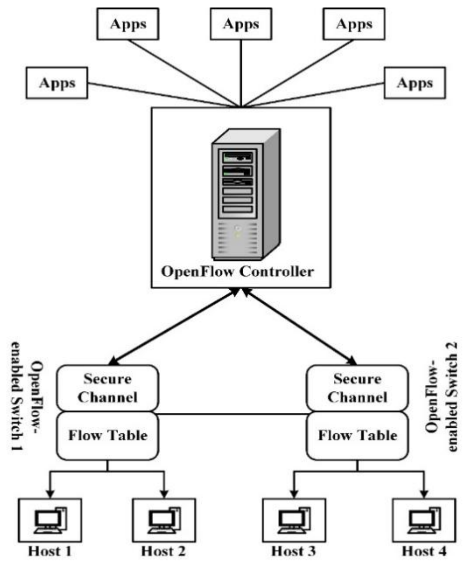

Figure 3. OpenFlow-based Network Framework

Figure 3 clearly illustrates the 3 phases of the OpenFlow-based network architecture that were earlier addressed. An intermediary layer between both the data plane and the application plane is a control layer made up of NOS. Once again, network administrators at end user machines define the network application. The switch in the illustration has a secure communication, a flow database, a grouping table, and metre tables, among other things. The switching description is laid out comprehensively. The controllers is in charge of managing all information flows entering an OpenFlow-enabled switch's ingress port by adding appropriate incoming packets to the switch's flow table, and the switch is only in charge of transmitting information when instructed by means of an OpenFlow controller. Every time a data packet approaches the switch, the switch begins comparing the header field of the arriving packet with the incoming packets listed in its forwarding devices.

If a match is made, the required flows procedures are carried out, and the information packets are formatted for delivery to their intended destination. If a match is not made, the switch would send the received packet to the following flow-table in a pipeline procedure. Once the last flow-table has received, the procedure is repeated. If there is still no matching, the switch would either send the newly arriving packet to the controllers or it might drop the packets in accordance with the controller's instructions. The pipeline processing's flow diagram and operation of several forwarding table are covered.

3.3. Vulnerability Evaluations of SDN

A secured communications infrastructure must have the following fundamental characteristics: security, stability, availability of information, authorization, and non-repudiation. Security experts should safeguard the information, the network resources, as well as the communication activities taking place over the internet in order to establish a system that is secure from deliberate assault or inadvertent loss. In order to maintain information security, it is necessary to evaluate the changes that SDN has made to the design of the system. Systems are getting more and more complicated. Hence, the probability and severity of vulnerabilities and operational faults have enhanced significantly to this intricacy. As more and more traditionally hardware-based operations are fulfilled by software, this complexity rises with network virtualisation. This significantly increases the burden on developers to produce faultless technology solutions. Furthermore, this demand is growing as a result of the trend in enterprise towards cloud technology. Network security is a priority in order to prevent harmful attackers from attacking them. Software-Defined Networking architecture-based networks must safeguard a variety of important security aspects, including:

- *Availability* – Even in the case of an assault, the system would continue to function.
- *Performance* – In the case of an intrusion, the network ought to have the capacity to ensure minimal throughput and delay.
- *Reliability and Discretion* – Tenants should maintain network control plane and data plane separation and authenticity.

There are several systems that must be in existence to guarantee the safeguarding of these resources. These are repudiation, multi-domain isolates, resilience, verification and authorisation, and durability. The procedures required to locate an unauthorised origin and then decide its access credentials are verification and authorisation.

When properly designed, these procedures could defend networks against threats like giving the platform misleading feedback, changing a legitimate on-path requests, forwarding traffic that was intended to be sent or not transmitting, and getting access to any component's management. The Software-Defined Networking must be protected with enhanced security precautions due to its crucial nature. As a result, reciprocal authentication is required for any communication within the control and data planes. Authentication process, along with security against replaying threats, secrecy, and stability assurance, are all possible with the help of security procedures like Transport layer security and IPSEC. From a security perspective, encryption and integrity protection without mutual verification are not effective. If there isn't already a widely recognized 3rd entity, the issue with mutual verification is that it necessitates prior knowledge of the remote communication destination. The main security risks include bandwidth flooding, denial of service attacks, susceptibility flooding, and connection flooding. Because of these security risks, attention must be drawn to the security considerations of Software-Defined Networking, the most recent emergent technologies.

4. Results and Discussion

4.1. Experimental Findings

To evaluate the performance of SDN, numerous simulation tools, such OMNET++ and Mininet, have been created. Other modelling tools include Ns-3 and Ns-2 Simulator was employed to execute the concepts presented in this study. The definition, architecture, advantages, and difficulties of Software-Defined Network and traditional networks were compared in this article. Together with every obstacle, such as Availability, Maliciousness, Packet duplication ratio, and Packet loss ratio of the both networking framework concept was also assessed. A comparison of the Software Defined Networks with traditional Networks and their associated parameters is performed, and a brief description of the graphical representation is presented below.

Availability

The percentage of uptime in a network structure over a given time frame is referred to as network availability. The amount of time a network is fully functional is referred to as uptime. The proportion of network availability is tracked to make sure the network functions reliably for users.

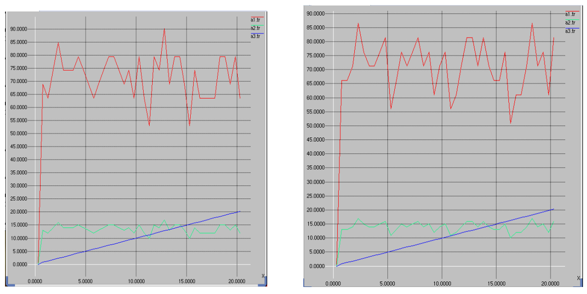
Figure 4. Graphical Representation of WSN-Availability and SDN-Availability

Figure 4 represents the network availability of WSN and SDN. When compared to traditional network, the graphical implementation result shows that the SDN network outperformed based on availability.

Malicious

Communications on a network could be seriously affected by a malevolent or malfunctioning of Networking device. Particularly, collaborating malevolent switches that attempt to conceal their improper behaviour are more difficult to find.

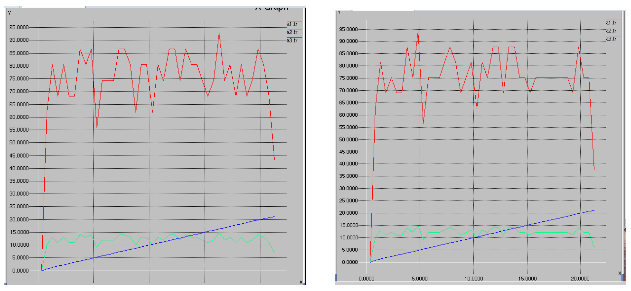
Figure 5. Graphical representation of WSN-malicious and SDN-malicious

Figure 4 represents the network anomalies of WSN and SDN. When compared to traditional network, the graphical implementation outcome illustrates that the SDN network outperformed based on Malicious node recognition.

Packet Duplication

When the same traffic is reported more than once as it moves across switch interfaces, this is referred to as packet duplication. Duplication can be caused by a number of port mirroring arrangements. These statistics that are gathered may be skewed by the existence of duplicate packets. Due to the fact that duplicate packets are treated as retransmissions, packet loss metrics are impacted.

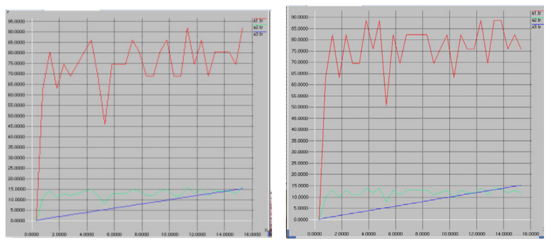

Figure 6. Graphical representation of WSN-packet duplication and SDN-packet duplication

Figure 6 represents the network packet duplication ratio of WSN and SDN. When compared to traditional network, the graphical implementation outcome illustrates that the SDN network outperformed based on Packet Duplication.

Packet Loss

In a computer network, packet loss happens when one or more data packets attempt to reach their destination but are unsuccessful. Network congestion or data transmission errors, which frequently occur when using wireless networks, are the two main causes of packet loss. By dividing the quantity of packets lost by the quantity of packets sent, packet loss is calculated.

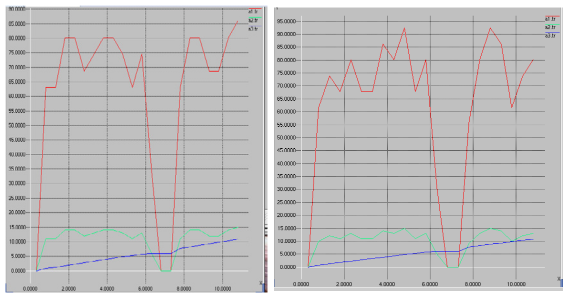

Figure 7. Graphical representation of WSN-packet loss and SDN-packet loss

Figure 7 represents the network packet loss ratio of WSN and SDN. When compared to traditional network, the graphical implementation outcome illustrates that the SDN network outperformed based on Packet Loss ratio.

Receiver

An electronic device that could receive information from a network is referred to as a receiver, such as a computer. Sender and receiver in communication networks are referred to as the nodes of a network. They are routed through network with each interconnected using switches or routers to perform packet transmission after it reaches their destination they are reassembled into the original data.

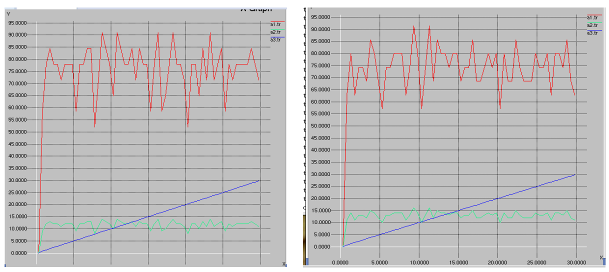

Figure 8. Graphical representation of WSN-receiver and SDN-receiver

Figure 8 represents the network receiver of WSN and SDN. When compared to traditional network, the graphical implementation outcome illustrates that the SDN networks outperformed based on receiver ratio..

5. Discussion

Traditional networks are difficult to administer and seem to be sophisticated. The majority of the causes for this are vertical integration and manufacturer-specific information and control planes. Software-Defined Networking provided an opportunity to overcome these persistent issues by splitting the Data plane from the Control plane, enhancing network flexibility, and standardising the control networks.Because to this SDN and its application were the subject of several studies rather than conventional networking. According to the analysis of the field, every study of Software-Defined Networking concentrated on a distinct aspect. This example demonstrated that, in contrast to traditional block chains, software-defined Internet of Things can ensure the privacy and stability of the networks utilising a secure and optimal power architecture [31]. The researchers demonstrated that networks' packet loss is reduced and their capacity is increased when the Software-Defined Networking technique is employed [32]. The research offered a comprehensive and reliable SDN architecture that could avoid Denial of service and spoofing threats with minimal SDN routers deployment costs [33]. The reference suggested TEDR techniques that could ensure optimal connections utilization when the nodes are installed as 30percent as entire and have a negligible impact on routing performance [34]. The studies illustrate that edge-based management on a centralised SDN controller might handle much more network traffic while retaining minimal delay. According to the literature, utilising the MCBLB technique in a architecture could boost load balancing by up to fourteen percent [35]. The researchers covered a method for preventing ARP spoofing that just involved adding a component to the Software-Defined Network controller, not any new hardware or software.

It was made clear that employing Flow technologies included into the controller demonstrates that the technique may identify and lessen DDoS assaults. According to the research, Software-Defined Networking offers a method that makes it simple to alter and re-program the information plane utilizing forwarding rules. Researchers created an optimisation structure and related flow design techniques which cut setup time in half and the duration required to restore interrupted processes in half, respectively [36]. The demonstrated Flow Keeper could sustain more than 80 percent of the overall of bandwidth during Assaults and it could stop unauthorised topological modifications by filtering out counterfeit link layer discovery protocol packets. The researchers investigated the efficiency and effectiveness of Phish Limiter as a method of phishing assault detection and prevention in Software-Defined Networking[37]. The article demonstrates how the unit price for the operation and the management adaptability of the design inversely correlated, with higher control scalability leading to a reduced unit service expenses [38]. According to the study [39], the efficiency of the Software Defined Networking controller was 32 percent less susceptible to DoS assaults when SDN-Guard was in operation. The researchers [40] demonstrated that big data and SDN could collaborate to develop an effective strategy for networking big data.

6. Conclusion

An innovative idea for managing and configuring computer networks is called "Software-Defined Networking." The primary goal of the SDN approach is to centralise network management at the SDN controller and decouple it from network switches. Instead of having to individually configure thousands of units, the centralised administration introduces an innovative method for managing network functions through a single application. The Software-defined networking idea is specifically discussed in this study with regard to certain parameters. The need for security upgrades is examined, and security-enhancing ideas are put forth. Yet, additional investigation is needed to create a highly secure system. The suggested approach was created utilizing the NS2 programme after researching the significant factors. A system for authentication is offered, securing communication among nodes and offering privacy. A Software-Defined Network,The potential technology for the future of the Internet and NGN is network depending on OpenFlow technologies.

Because data and control planes are vertically integrated on network core components, conventional networks have complicated designs and could be challenging to administer. Owing to the fact that the A Software-Defined Network architecture separates the data plane and control plane from the network key devices, the vertical integration issue that plagues traditional networks is addressed. By taking into account different networking topologies, this article examines the efficiency of traditional networks versus networks that support OpenFlow. The effectiveness of a suggested scheme is compared among a conventional network environment as well as an OpenFlow-enabled network environment. Based on the findings, it could be said that networks with OpenFlow support perform better than conventional networks in real-time environments.

References:

[1]. Lichtwald, G., Walter, U., & Zitterbart, M. (2004). Improving convergence time of routing protocols. In *Proceedings of 3 International Conference on Networking (ICN'04)*.

[2]. Zhang, H., & Yan, J. (2015). Performance of SDN routing in comparison with legacy routing protocols. In *2015 International conference on cyber-enabled distributed computing and knowledge discovery*, 491-494. IEEE.

[3]. Sankar, D., & Lancaster, D. (2013). Routing protocol convergence comparison using simulation and real equipment. *Advances in Communications, Computing, Networks and Security*, *10*, 186-194.

[4]. Panford, J.K, Riverson, K, Oliver, B.K & Yehuza, R.M (2015). Comparative Analysis Of Convergence Times Between RIP And EIGRP Routing Protocols In A Network. Res. J. Comput. Sci., *2*(3), 1–10.

[5]. Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14-76.

[6]. Paul, S., Pan, J., & Jain, R. (2011). Architectures for the future networks and the next generation Internet: A survey. *Computer Communications*, *34*(1), 2-42.

[7]. Gavras, A., Karila, A., Fdida, S., May, M., & Potts, M. (2007). Future internet research and experimentation: the FIRE initiative. *ACM SIGCOMM Computer Communication Review*, *37*(3), 89-92.

[8]. Pan, J., Paul, S., & Jain, R. (2011). A survey of the research on future internet architectures. *IEEE Communications Magazine*, *49*(7), 26-36.

[9]. Jammal, M., Singh, T., Shami, A., Asal, R., & Li, Y. (2014). Software defined networking: State of the art and research challenges. *Computer Networks*, *72*, 74-98.

[10]. Lara, A., Kolasani, A., & Ramamurthy, B. (2013). Network innovation using openflow: A survey. *IEEE communications surveys & tutorials*, *16*(1), 493-512.

[11]. Zebari, R. R., Zeebaree, S. R., Jacksi, K., & Shukur, H. M. (2019). E-business requirements for flexibility and implementation enterprise system: A review. *International Journal of Scientific & Technology Research*, *8*(11), 655-660.

[12]. Zeebaree, S., Ameen, S., & Sadeeq, M. (2020). Social media networks security threats, risks and recommendation: A case study in the kurdistan region. *International Journal of Innovation, Creativity and Change*, *13*(7), 349-365.

[13]. Alzakholi, O., Shukur, H., Zebari, R., Abas, S., & Sadeeq, M. (2020). Comparison among cloud technologies and cloud performance. *Journal of Applied Science and Technology Trends*, *1*(2), 40-47.

[14]. Xu, H., Huang, H., Chen, S., Zhao, G., & Huang, L. (2018). Achieving high scalability through hybrid switching in software-defined networking. *IEEE/ACM Transactions on Networking*, *26*(1), 618-632.

[15]. Kareem, F. Q., Zeebaree, S. R., Dino, H. I., M Sadeeq, M. A., Rashid, Z. N., Hasan, D. A., & Sharif, K. H. (2021). A survey of optical fiber communications: challenges and processing time influences. *Asian Journal of Research in Computer Science*, *7*(4), 48-58.

[16]. Zebari, I. M., Zeebaree, S. R., & Yasin, H. M. (2019). Real time video streaming from multi-source using client-server for video distribution. In *2019 4th Scientific International Conference Najaf (SICN)*, 109-114. IEEE.

[17]. Zeebaree, S. R., Shukur, H. M., Haji, L. M., Zebari, R. R., Jacksi, K., & Abas, S. M. (2020). Characteristics and analysis of hadoop distributed systems. *Technology Reports of Kansai University*, *62*(4), 1555-1564.

[18]. Ageed, Z. S., Zeebaree, S. R., Sadeeq, M. M., Kak, S. F., Yahia, H. S., Mahmood, M. R., & Ibrahim, I. M. (2021). Comprehensive survey of big data mining approaches in cloud systems. *Qubahan Academic Journal*, *1*(2), 29-38.

[19]. Perepelkin, D., & Tsyganov, I. (2019, October). SDN cluster constructor: software toolkit for structures segmentation of software defined networks. In *2019 XVI International Symposium" Problems of Redundancy in Information and Control Systems"(REDUNDANCY)*, 195-198. IEEE.

[20]. Abdulqadir, H. R., Zeebaree, S. R., Shukur, H. M., Sadeeq, M. M., Salim, B. W., Salih, A. A., & Kak, S. F. (2021). A study of moving from cloud computing to fog computing. *Qubahan Academic Journal*, *1*(2), 60-70.

[21]. Shukur, H., Zeebaree, S. R., Ahmed, A. J., Zebari, R. R., Ahmed, O., Tahir, B. S. A., & Sadeeq, M. A. (2020). A state of art survey for concurrent computation and clustering of parallel computing for distributed systems. *Journal of Applied Science and Technology Trends*, *1*(4), 148-154.

[22]. Dino, H. I., Zeebaree, S. R., Ahmad, O. M., Shukur, H. M., Zebari, R. R., & Haji, L. M. (2020). Impact of load sharing on performance of distributed systems computations. *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, *3*(1), 30-37.

[23]. Yahia, H. S., Zeebaree, S. R., Sadeeq, M. A., Salim, N. O., Kak, S. F., AL-Zebari, A., ... & Hussein, H. A. (2021). Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. *Asian Journal of Research in Computer Science*, *8*(2), 1-16.

[24]. Manshahia, M. S. (2016). Wireless sensor networks: a survey. *International Journal of Scientific & Engineering Research*, *7*(4), 710-716.

[25]. Chen, D., & Varshney, P. K. (2004, June). QoS support in wireless sensor networks: a survey. In *International conference on wireless networks*, *233*, 1-7.

[26]. Zou, L., Lu, M., & Xiong, Z. (2005). A distributed algorithm for the dead end problem of location based routing in sensor networks. *IEEE transactions on vehicular technology*, *54*(4), 1509-1522.

[27]. Rawat, D. B., & Reddy, S. R. (2016). Software defined networking architecture, security and energy efficiency: A survey. *IEEE Communications Surveys & Tutorials*, *19*(1), 325-346.

[28]. Abd Elazim, N. M., Sobh, M. A., & Bahaa-Eldin, A. M. (2018). Software defined networking: attacks and countermeasures. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 555-567. IEEE.

[29]. Bholebawa, I. Z., & Dalal, U. D. (2018). Performance analysis of SDN/OpenFlow controllers: POX versus floodlight. *Wireless Personal Communications*, *98*, 1679-1699.

[30]. Bholebawa, I. Z., Jha, R. K., & Dalal, U. D. (2016). Performance analysis of proposed OpenFlow-based network architecture using mininet. *Wireless Personal Communications*, *86*, 943-958.

[31]. Rahman, A., Islam, M. J., Montieri, A., Nasir, M. K., Reza, M. M., Band, S. S., ... & Mosavi, A. (2021). Smartblock-sdn: An optimized blockchain-sdn framework for resource management in iot. *IEEE Access*, *9*, 28361-28376.

[32]. Vishnevsky, V., Kirichek, R., Elagin, V., Vladyko, A., & Shestakov, A. (2020). SDN-assisted unmanned aerial system for monitoring sensor data. In *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT),* 313-317. IEEE.

[33]. Ruaro, M., Caimi, L. L., & Moraes, F. G. (2020). A systemic and secure SDN framework for NoC-based many-cores. *IEEE Access*, *8*, 105997-106008.

[34]. Ren, C., Bai, S., Wang, Y., & Li, Y. (2020). Achieving near-optimal traffic engineering using a distributed algorithm in hybrid SDN. *Ieee Access*, *8*, 29111-29124.

[35]. Al-Tam, F., & Correia, N. (2019). On load balancing via switch migration in software-defined networking. *IEEE Access*, *7*, 95998-96010.

[36]. Achleitner, S., Bartolini, N., He, T., La Porta, T., & Tootaghaj, D. Z. (2018). Fast network configuration in software defined networking. *IEEE Transactions on Network and Service Management*, *15*(4), 1249-1263.

[37]. Chin, T., Xiong, K., & Hu, C. (2018). Phishlimiter: A phishing detection and mitigation approach using software-defined networking. *IEEE Access*, *6*, 42516-42531.

[38]. Karakus, M., & Durresi, A. (2017). Service cost in software defined networking (SDN). In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 468-475. IEEE.

[39]. Dridi, L., & Zhani, M. F. (2016). SDN-guard: DoS attacks mitigation in SDN networks. In *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*, 212-217. IEEE.

[40]. Cui, L., Yu, F. R., & Yan, Q. (2016). When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE network*, *30*(1), 58-65.