A Comparative Study of the Energy Efficiency of Traditional Network Topology and Software-Defined Networking (SDN) Topology

Sara Hameed[1], Usra Sami[2], Sara Naeem Aslam[1], Meerab Tahir[1]
[1]Software Engineering Department, Ned University of Engineering and Technology, Pakistan
[2] Computer Engineering Department, Bahria University Karachi, Pakistan
*Correspondence: hameed4402051@cloud.neduet.edu.pk, usrasami.bukc@bahria.edu.pk, aslam4406044@cloud.neduet.edu.pk, tahir4405501@cloud.neduet.edu.pk

Citation | Hameed. S, Sami. U, Aslam. S. N Tahir. M, "A Comparative Study of the Energy Efficiency of Traditional Network Topology and Software-Defined Networking (SDN) Topology", IJIST, Vol. 7, Issue. 3 pp 1341-1352, July 2025
Received | June 3, 2025 **Revised |** June 26, 2025 **Accepted |** June 30, 2025 **Published |** July 7, 2025.

This study is intended to be a comparative study of the energy efficiency of the traditional network topology and the software-defined networking (SDN) topology. Energy efficiency has been a priority aspect in network design due to environmental concerns and cost optimization of operational costs. Traditional networking is based on the existing configuration of the hardware devices and decentralized control, which in turn results in ineffective usage of the resources. However, in contrast, SDN centralizes network control and thus facilitates energy-efficient resource allocation and optimization. In this comparison, the energy consumption profiles, the utilization patterns of the resources, and the operating strategies of both methods are evaluated. The goal of this study is to present information on SDN energy efficiency over the conventional networking method and to demonstrate how SDN can offer benefits to the environment and economy by using SDN technologies in network infrastructures. However, the selection of the specific network infrastructure may vary depending on specific user requirements.

Keywords: Software Defined Networking, Traditional Networking, Energy Efficiency, Networking

Introduction:

In this technological era of development and the escalating needs for interconnectedness, the need for efficient network architecture cannot be neglected. Thus, when analyzing existing networks as well, their constantly increasing size and variety lead to more energy consumption. Without implementing efficient energy solutions, we face several critical issues such as an overall upsurge in operational costs, amplified environmental footprint, and enhanced detrimental effects. Unproductive networks not only ruin capital but also resources, much needed to combat the disastrous effects of global warming and environmental depletion [1].

In the past, communication networks have been designed assuming that the observed traffic will be the same throughout busy hours, which causes many networks to over-provision during off-peak hours. This has led to many researchers being dedicated to providing energy-conserving measures, which include switching off or setting parts of the network into sleep mode without compromising its functionality. Therefore, it is important to find a way out to make it energy efficient.

Furthermore, one of the ways of attaining this challenge is by adopting Software Defined Networks. SDN is an architectural approach where the data plane and the control plane are not in the same. Whereas, traditional networks have both data plane and control plane in the same network, and this does not allow the priorities to be changed quickly, and it also makes managing the network more complex. This is in contrast to SDN, where the control plane is an independent central controller while the data plane is in the devices in a network. Such separation allows for centralization and flexibility in the management of the organization's human resource divisions [2].

The SDN controller achieves global knowledge of the network, and as such, it controls all the resources in the network and the flow of packets. When the controller is empowered to forward, filter, and otherwise manage traffic through the network, and does so in a way that optimizes the use of network components. Therefore, SDN makes it energy efficient [3].

In SDN, the network elements are capable of conveying data to a controller through protocols that include OpenFlow. According to the instructions issued by the controller, the devices themselves control the flow of traffic and data trans- mission. This centralized control helps automate, decreases time to reconfigure the network, and includes support for power-saving measures such as powering down unnecessary links or putting non-active devices to sleep.

In addition to this, SDN can bring fundamental improvements in energy efficiency and concurrently provide similar levels of efficiency as those provided by current conventional networks. For example, while controlling the interactions of the components with minimal utilization through SDN, it would be possible to deactivate these components and thereby cut down on energy and operational costs. Moreover, the traditional branching and calculation capabilities of the SDN provide a constant network resource reuse to maintain optimum utilization of the available means and prevent the growth of energy consumption [4].

Therefore, a comparative analysis has been done between the traditional network and software design networks. Practical implementation has been done on the Mininet simulator. This has been done by replacing conventional routers with an SDN controller. Link utilization is done by the controller, and network resources are controlled to ensure minimal energy use at any given time.

It is essential to switch to energy-efficient networks for economic benefits as well as for their impact on the environment. However, SDN is useful for dynamic control over the resources gathered across the network without any need to deploy additional hardware, thus ensuring greater energy efficiency and therefore a lower negative impact on the environment for large-scale network infrastructures. The significance of this study will focus on identifying the viability of SDN in preventing the communication network from becoming unsustainable and expensive by analyzing its ability to minimize energy use with the help of smart resource control. Energy efficiency in networks can be achieved through the use of smart hardware, but in this research, we have enhanced energy efficiency in traditional topology by reducing CPU and memory usage [5].

Agg et al. [6] present the architecture of SDN networks, as illustrated in Figure 1.

- Analyses the energy efficiency implications by observing the resources consumed when operating a network.
- To identify the difference in scalability and manageability between traditional and SDN network topologies.
- To be able to simulate the two styles of topology, the traditional and the SDN, under similar conditions, and most specifically Star topologies using Mininet.

Novelty Statement:

Although many of the published works have discussed the hypothetical benefits of Software-Defined Networking (SDN) over conventional network designs, the study will provide us with a distinct and empirical type of comparison work on the two types of networks based on the empirical base generated through simulation in Mininet. Our research, unlike other studies that tend to deal with qualitative evaluation or unrelated scores, covers a wider picture with the set of key performance indicators such as latency, throughput, CPU and memory utilization, and packet loss rate. This study points towards the practical advantage of SDN by virtualizing the conventional and the SDN-enabled star network in an identical environment and showing the enhanced energy efficiency, significant reductions in computational overhead, and better bandwidth utilization. Moreover, the research also highlights the practical aspects of the centralized approach in SDN since it can make configuration easier and more scalable. This work is novel in the fact that, besides performance measurement, the presented performance measurement and practical resource analysis complement each other to provide a better insight into the SDN benefits in contemporary network infrastructures.

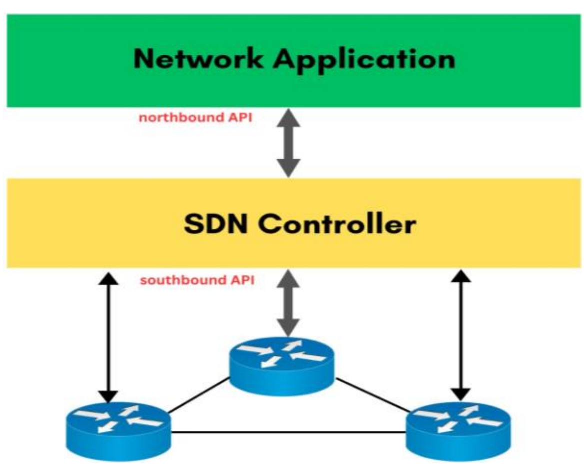

Figure 1. SDN, Architecture

Objectives:

This research aims at achieving the following main objectives:

- To examine and compare the performance of Traditional Networks and Software-Defined Networking (SDN) regarding computational resource consumption (CPU and memory usage), latency, packet loss, and throughput, and discuss the results.

Literature Review:

With the advent of this fast-paced, dynamic world, the need for efficient networks has emerged. [7] As a result, SDN has emerged with the concept of making the traditional networks much more efficient in different aspects, and it has further helped in energy efficiency.

Saad Himmat etal in 2021 in their research paper "The comparison of Software Defined networking with traditional networking" have discussed that SDN is important. It can easily manage and track network operations because it handles forwarding and routing differently. [8]

Similarly, Mousa M etal in 2016 in their research have described that the major difference between traditional networks and SDN is that SDN is a better approach because it deals with network operations remotely through software, whereas traditional networks require different physical devices and hardware connections. [5]

Xu et al. in 2018 have discussed that because of the ease of network management in SDN, it has become easier to use and versatile as compared to traditional networks, which are complex and difficult [9]

Sufiev H in 2016 explained that the traditional networks are more complex because different routers, physical devices, and switches are connected, which is not easy to manage. [10]

On the other hand, Yazdeen etal in 2021 in their research, and AlShehri in 2017 have found out that in conventional networks, the data planes are in a single unit, which increases the load on CPU and memory and affects the reliability. Whereas, SDN manages these processes separately and controls the load and traffic flow. [11][12]

Furthermore, Netes V in 2019 and Rawat DB in 2017 have claimed that SDN does not require additional cost since it does not require any additional connections, whereas traditional networks require a lot of cost for physical networks [13][14]. Irena Seremet etal in 2019 have described different studies in their research paper "SDN as a Tool for energy saving" [15]

The first literature discussed by Gelenbe in 2009 has been a reduction in the usage of energy through delay and packet loss. It has been found that more than 30 percent of the energy consumption has been reduced. Whereas, by implementing this, it has been noticed that there is packet loss and the transmission results in unexpected delays [16].

R Carpa in 2014 decided to use an intra-domain SDN solution, which resulted in better quality and a reduction in link consumption. [17]

Radu Carpa, in 2017, in the research "Energy Efficient Traffic Engineering in Software Defined Networks," described that the energy process was optimized through a single operator domain. This was implemented using the ONOS SDN controller along with SDN switches. [18]

Furthermore, F. A. Moghaddam in 2016 described four types of algorithms to improve the power savings in the SDNs. These measures have advanced, including setting inactive devices to sleep state, setting inwoke switches over sleep switches, and augmenting the quantity of sleeping switches. [19]

Similarly, Kra L etal in 2018, in [20], have explained that SDN technology has been applied to make Links dormant when not in use to avoid adhering to congestion for the purpose of quality. This approach has proven to be reasonable in understanding that the rate might be as low as 33. 33 percent of links. Technological advancement: Ports have agreed with the notion that 5 percent of ports could have been shut down to save energy. [20]

A survey carried out by Tuysuz in 2017 has shown that most of the solutions entailed simply redirecting flows through the network to reduce power consumption by eliminating many switches. [21]

These studies have put on display the potential that lies in SDN in increasing energy efficiency in networks. Irena Seremet, in the work published in 2009, analyzed SDN as the means to save energy [15] while Peter Andra´s Agg and Zsolt Csaba Johanya´kin have done the same in their 2021 publication. They remain highly energy-intensive and prove unsustainable to the environment and unaffordable for the end user. [6]

Methodology:

Our research is based on the consideration of a star topology, where each node is connected to a central hub or switch.

Communication between nodes occurs through this hub or switch. For our research, we utilized PuTTY and the Xming server to access the Mininet GUI, specifically the 'miniedit.py 'tool. This setup allowed us to effectively manage and visualize network topologies within our study.

Setup Topologies:

Set up the Traditional Star Topology: According to [22], the traditional star topology consists of a single switch connected to multiple hosts, operating without any centralized controller.

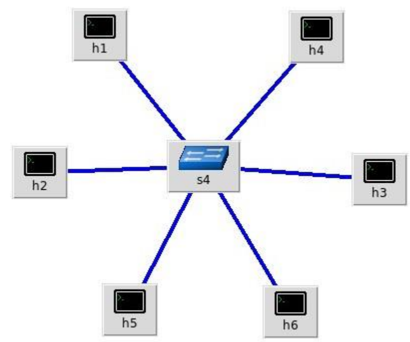

Figure 2. Traditional Star Topology

Figure 2 illustrates the traditional star topology, which consists of 6 hosts connected to 1 switch.

Set up the SDN Star Topology: According to [22], the SDN network consists of a single OpenFlow switch connected to multiple hosts. The OpenFlow switch is linked to an OpenFlow controller through a secure channel.

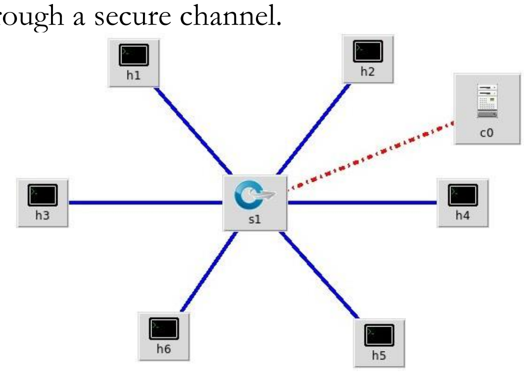

Figure 3. SDN Star Topology

Figure 3 illustrates the SDN star topology, having 6 hosts connected to a single switch, with the switch connected to a controller.

Run Ping Tests:

Run the ping command to establish communication between each node and verify traffic flow.

Traditional Star Topology: The 'pingall 'command is used to check connectivity as shown in Fig. 4.

```
mininet> pingall
*** Ping: testing ping reachability
h3 -> h6 h4 h1 h5 h2
h6 -> h3 h4 h1 h5 h2
h4 -> h3 h6 h1 h5 h2
h1 -> h3 h6 h4 h5 h2
h5 -> h3 h6 h4 h1 h2
h2 -> h3 h6 h4 h1 h5
*** Results: 0% dropped (30/30 received)
```

Figure 4. Pinging all nodes

SDN Star Topology: The 'pingall 'command is used to check connectivity, as shown in Fig. 5.

```
mininet> pingall
*** Ping: testing ping reachability
h3 -> h6 h4 h1 h5 h2
h6 -> h3 h4 h1 h5 h2
h4 -> h3 h6 h1 h5 h2
h1 -> h3 h6 h4 h5 h2
h5 -> h3 h6 h4 h1 h2
h2 -> h3 h6 h4 h1 h5
*** Results: 0% dropped (30/30 received)
```

Figure 5. Pinging all nodes

Performance metrics:

The metrics we employ to compare traditional and SDN network topologies, as part of our methodology, are listed below.

1. Latency: The time delay between sending a packet from the source to receiving it at the destination is calculated.

mininet> h1 ping -c 10 h1

2. Measure Throughput: Use the iperf command to measure network throughput. Start the server on one host and run the client on another host for 60 seconds.

mininet> h1 iperf -s & mininet> h2 iperf -c h1 -t 60

3. Monitor CPU and Memory Usage: On a host, the 'top 'command is used to monitor CPU and memory usage.

mininet> h1 top &

4. Network Analysis: Capture network traffic using 'tcpdump '.

mininet> h1 tcpdump -i h1-eth0 -w capture.pcap

5. Check Memory Usage: Use 'cat /proc/meminfo 'to check memory usage.

mininet> h1 cat /proc/meminfo

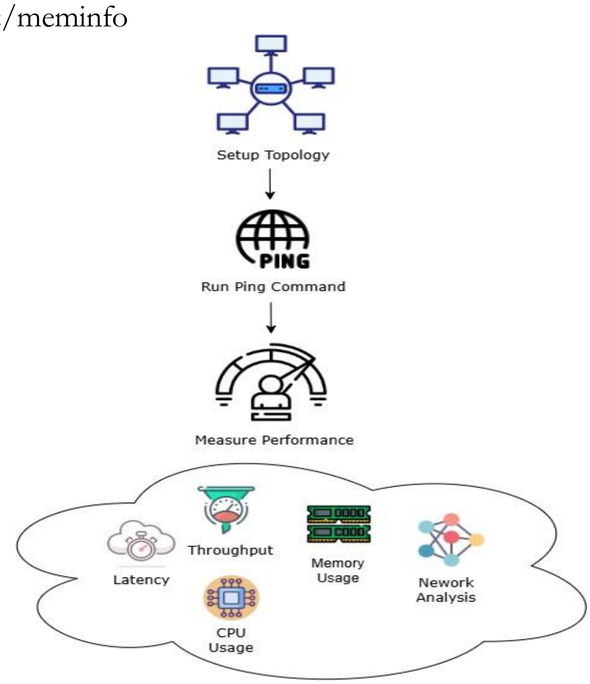

Figure 6. Process Flow of the Research Methodology

Figure 6 illustrates the overall methodology steps, which include setting up the network topology, executing the ping command, and measuring performance metrics such as latency, throughput, memory usage, CPU usage, and network analysis.

Comparative Analysis:

To demonstrate that an SDN (Software-Defined Network- ing) topology is more efficient than a traditional topology, we conduct a comparative analysis based on metrics collected from both setups. Key metrics such as latency, throughput, scalability, and ease of management were significantly improved in the SDN environment compared to the traditional setup. This comparison underscores the advantages of adopting SDN, highlighting its capability to optimize network performance and operational efficiency.

Ping Latency Performance and Packet Loss Rate:

1) **Traditional Star Topology:**

- Minimum RTT: 0.052 ms
- Average RTT: 0.123 ms
- Maximum RTT: 0.271 ms
- Packet Loss: 0

Figure 7 illustrates the latency and packet loss observed in a traditional network topology.

```
mininet> h2 ping -c 10 h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.271 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.082 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.096 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.090 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.128 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.141 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.080 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.100 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.190 ms

--- 10.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8999ms
rtt min/avg/max/mdev = 0.052/0.123/0.271/0.061 ms
```

Figure 7. Latency and Packet Loss Rate in Traditional Topology

2) **SDN Star Topology:**

- Minimum RTT: 0.075 ms
- Average RTT: 0.398 ms
- Maximum RTT: 1.699 ms
- Packet Loss: 0

Figure 8 illustrates the latency and packet loss observed in an SDN network topology.

```
mininet> h2 ping -c 10 h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.13 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.399 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.165 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.126 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.075 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.080 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.083 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.109 ms

--- 10.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9001ms
rtt min/avg/max/mdev = 0.075/0.398/1.699/0.533 ms
```

Figure 8. Latency and Packet Loss Rate in SDN Topology

The traditional topology shows lower minimum and average RTTs, indicating faster ping responses. However, the maximum RTT is slightly higher in the SDN topology, suggesting more variation in response times. The packet loss is 0 percent in both cases, indicating reliable communication.

Network Throughput (Iperf Test):

Network throughput is analyzed to determine which topology, traditional or SDN, handles data more efficiently.

3) **Traditional Star Topology:**

- Bandwidth: 17.1 Gbits/sec

4) **SDN Star Topology:**

- Bandwidth: 19.4 Gbits/sec

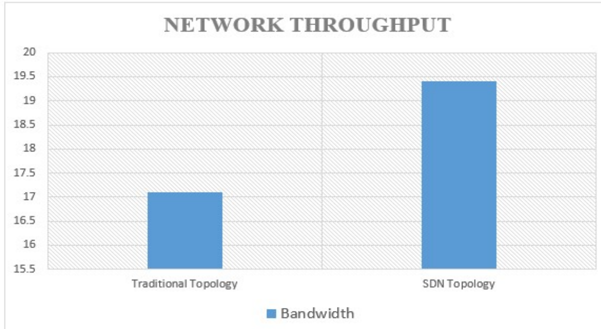

Figure 9. Comparison between the Bandwidth of traditional and SDN topology

The graph in Figure 9 demonstrates that the SDN topology provides greater bandwidth compared to the traditional topology, indicating that the network can handle data transfer more efficiently.

CPU and Memory Usage:

The CPU and memory usage are compared for both network topologies, traditional and SDN, to evaluate their efficiency and performance. This comparison is done to identify the effective utilization of system resources across both topologies while handling network operations.

1) **Traditional Star Topology:**

- CPU Usage: 1.7 percent
- Memory Usage: 3.1 percent

```
top - 06:15:35 up 12 min,  3 users,  load average: 0.28, 0.40, 0.22
Tasks:  93 total,   1 running,  92 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem :  1023812 total,   790904 free,    86052 used,   146856 buff/cache
KiB Swap:   998396 total,   998396 free,        0 used.   784572 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1509 root      20   0       0      0      0 S  0.7  0.0   0:02.71 kworker/0:2
    6 root      20   0       0      0      0 S  0.3  0.0   0:01.29 kworker/u2:0
    7 root      20   0       0      0      0 S  0.3  0.0   0:00.74 rcu_sched
  871 root      10 -10   34048  32068   6300 S  0.3  3.1   0:01.52 ovs-vswitchd
 1043 mininet   20   0   10720   6156   5188 S  0.3  0.6   0:00.29 xterm
 1744 root      20   0    7956   3400   2984 R  0.3  0.3   0:00.10 top
    1 root      20   0    6484   4924   3800 S  0.0  0.5   0:04.06 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:00.50 ksoftirqd/0
    4 root      20   0       0      0      0 S  0.0  0.0   0:00.66 kworker/0:0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
   10 root      rt   0       0      0      0 S  0.0  0.0   0:00.08 watchdog/0
   11 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   12 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
   13 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 perf
```

Figure 10. CPU and Memory usage

2) **SDN Star Topology:**

- CPU Usage: 1.4 percent
- Memory Usage: 2.9 percent

```
top - 06:34:39 up 31 min,  3 users,  load average: 0.02, 0.14, 0.12
Tasks: 102 total,   1 running,  91 sleeping,   2 stopped,   8 zombie
%Cpu(s):  0.0 us,  0.7 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  1023812 total,   775956 free,    92064 used,   155792 buff/cache
KiB Swap:   998396 total,   998396 free,        0 used.   777844 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  819 ntp       20   0   15624   4224   3788 S  0.3  0.4   0:00.95 ntpd
  852 root      10 -10    5504   3768   3344 S  0.3  0.4   0:00.49 ovsdb-server
  871 root      10 -10   34048  32068   6300 S  0.3  3.1   0:04.69 ovs-vswitchd
  952 mininet   20   0   10756   3904   3136 S  0.3  0.4   0:10.19 sshd
 2075 root      20   0   14588   9856   6072 S  0.3  1.0   0:00.21 python
 2113 root      20   0       0      0      0 S  0.3  0.0   0:01.30 kworker/0:0
 2119 root      20   0       0      0      0 S  0.3  0.0   0:00.16 kworker/u2:3
 2366 root      20   0    7956   3324   2904 R  0.3  0.3   0:00.12 top
    1 root      20   0    6484   4924   3800 S  0.0  0.5   0:04.14 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:00.90 ksoftirqd/0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S  0.0  0.0   0:01.09 rcu_sched
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
   10 root      rt   0       0      0      0 S  0.0  0.0   0:00.20 watchdog/0
   11 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
```

Figure 11. CPU and Memory usage

Figures 10 and 11 depict CPU usage in both traditional and SDN topologies. The slightly reduced CPU usage in the SDN setup indicates that incorporating a controller helps lower overhead, thereby improving energy efficiency.

Result:

Analysis of the obtained results by the performance assessment (with Mininet simulation) showed significant disparities between a familiar network and Software-Defined Networking (SDN) in respect of latency, throughput, consuming resources, and efficacy in general. With network latency, the time-tested Star topology scored better with a lowest Round-Trip Time (RTT) of 0.052 milliseconds, an average RTT of 0.123 milliseconds, and a highest RTT of 0.271 milliseconds. In opposed, the SDN-based star topology showed a slightly longer average value of latency with minimal RTT of 0.075 milliseconds, an average RTT of 0.398 milliseconds, and a high RTT that extended to 1.699 milliseconds. The reason behind this latency increase in the SDN environment is that the SDN controller can take part in the decision-making process; hence, there is some degree of delay, especially during the time when the first packet forwarding is taking place. Still, irrespective of latency correction, neither of the network architectures lost any packets throughout the simulation, which is what qualifies reliability and robustness in the form of data transmission.

In the case of throughput, the SDN architecture was better than the traditional model. SDN star topology obtained a bandwidth of 19.4 Gbps as opposed to 17.1 Gbps in traditional star topology. This improved throughput of SDN can be attributed to the central control plane that allows a more efficient routing decision and optimized bandwidth utilization. This enhanced network performance is also because SDN has been able to dynamically manage traffic flows, particularly where the amount of data exchanged and the number of nodes are high. The second value of this research was resource use, that is, CPU and memory consumption, the indicators of energy efficiency. The analysis indicated that SDN, with its topology operation, uses less computing power. CPU utilization with SDN configuration was also very low at 1.4% as compared to the 1.7% in the conventional network. In the same way, SDN topology memory consumption was 2.9% as opposed to 3.1% in the usual setup. The findings show that SDN has the potential to enable energy-efficient network activities as the responsibilities of making decisions are transferred to a centralized controller, thus lessening the burden on individual switches and routers. SDN also has great benefits to the management and scalability of resources. Whereas the conventional networks face a system of manual configuration and troubleshooting techniques on every single device, the SDN can deploy a central policy establishment and automatic configuration with rapid deployment of modifications across the network. This saves not only administrative overhead, but also minimizes human error, plus the capacity to respond rapidly to dynamic network circumstances or failure. In general, the findings of the present study have a few strong points that suggest SDN not just improves bandwidth and decreases resource utilization, but has long-term advantages in dynamicity and cost-effectiveness of the network operation, and, therefore, it can become a rather interesting option to traditional networking in contemporary infrastructures.

Table 1. Metrics Results

Performance Metrics	Traditional Topology	SDN Topology
Packet Loss	0	0
Latency Bandwidth	17.1 Gbit per sec	19.4 Gbit per sec
CPU Usage	1.7 percent	1.4 percent
Memory Usage	3.1 percent	2.9 percent

Table 1 presents the results of a comparative analysis of the traditional topology and SDN topology.

Discussion:

The findings of this comparison analysis show that Software-Defined Networking (SDN) topology has significant advantages over traditional network design in terms of energy efficiency and overall network performance. The study clearly shows that SDN beats traditional networks in key critical measures, including latency, throughput, system resource con- consumption, scalability, and simplicity of maintenance.

One of the most notable findings is the reduction in latency in SDN systems. By decoupling the control plane from the data plane and centralizing control, SDN decreases the delays normally found in conventional networks. Similarly, throughput improvements in SDN can be ascribed to the network's capacity to dynamically change traffic flows in response to current demands. This adaptability guarantees that existing resources are optimally utilised and minimizes congestion.

Importantly, the study demonstrates that SDN uses fewer system resources, including CPU and memory. The centralized control paradigm enables more efficient decision-making and reduces redundant processing across network devices. In con- trast, traditional networks frequently require each device to control its routing. When considering scalability, SDN offers a more versatile infrastructure. SDN controllers' centralized structure allows for the addition of new devices and services with minimal human configuration, making them excellent for growing or developing network environments.

SDN's ease of management is another significant advantage. Network policies, updates, and security configurations may be centrally controlled and uniformly applied across the network, considerably lowering administrative costs and human error when compared to traditional solutions.

These findings align with previous research that emphasizes the transformative potential of SDN for developing smarter, greener, and more flexible networks. However, it is crucial to note that the first implementation of SDN may incur some costs and difficulties, especially for organizations switching from traditional systems. Overall, the findings support the notion that SDN topologies provide a more energy-efficient and performance-optimized solution than conventional network topologies.

Conclusion:

In SDN networks, energy optimization has become an important element, leading to lower resource usage and better outcomes. The centralized approach of the SDN OpenFlow controller has played a significant role in this regard. It is important to note that to effectively compare energy efficiency between traditional and SDN topologies, it's essential to consider a larger network, such as that of an entire workplace. These benefits are particularly valuable in environments where networks are complex and scalability and adaptability are crucial. We provide a thorough comparison between the traditional topology and SDN topology, focusing on the star topology configuration. Memory optimization, CPU consumption, latency, and throughput are significant aspects we considered in this research.

Despite the existence of some solutions addressing various dimensions of energy efficiency in SDN, there are still open research issues, challenges, and areas for improvement.

References:

[1] T. G. Akram Hakiri, Aniruddha Gokhale, Pascal Berthou, Douglas C. Schmidt, "Software-Defined Networking: Challenges and research opportunities for Future Internet," *Comput. Networks*, vol. 75, pp. 453–471, 2014, doi: https://doi.org/10.1016/j.comnet.2014.10.015.

[2] Sumit Badotra, "A Review Paper on Software Defined Networking," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 02, 2017, [Online]. Available: https://www.researchgate.net/publication/319550243_A_Review_Paper_on_Softwa

re_Defined_Networking
[3] Y. L. Manar Jammal, Taranpreet Singh, Abdallah Shami, Rasool Asal, “Software defined networking: State of the art and research challenges,” *Comput. Networks*, vol. 72, pp. 74–98, 2014, doi: https://doi.org/10.1016/j.comnet.2014.07.004.
[4] W. Zhong, R. Yu, S. Xie, Y. Zhang, and D. H. K. Tsang, “Software Defined Networking for Flexible and Green Energy Internet,” *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 68–75, Dec. 2016, doi: 10.1109/MCOM.2016.1600352CM.
[5] M. Mousa, A. M. Bahaa-ElDin, and M. Sobh, “Software Defined Networking concepts and challenges,” *Proc. 2016 11th Int. Conf. Comput. Eng. Syst. ICCES 2016*, pp. 79–90, Jan. 2017, doi: 10.1109/ICCES.2016.7821979.
[6] Z. C. J. Péter András Agg, “Energy Savings In Sdn Networks,” *Gradus*, vol. 8, no. 1, pp. 205–210, 2021, [Online]. Available: https://gradus.kefo.hu/archive/2021-1/2021_1_CSC_005_Agg.pdf
[7] H. A. Eissa, K. A. Bozed, and H. Younis, “Software Defined Networking,” *19th Int. Conf. Sci. Tech. Autom. Control Comput. Eng. STA 2019*, no. March 2019, pp. 620–625, 2019, doi: 10.1109/STA.2019.8717234.
[8] H. M. Y. Saad H. Haji, Subhi R. M. Zeebaree, Rezgar Hasan Saeed, Siddeeq Y. Ameen, Hanan M. Shukur, Naaman Omar, Mohammed A. M. Sadeeq, Zainab Salih Ageed, Ibrahim Mahmood Ibrahim, “Comparison of Software Defined Networking with Traditional Networking,” *Asian J. Res. Comput. Sci.*, vol. 9, no. 2, pp. 1–18, 2021, doi: 10.9734/ajrcos/2021/v9i230216.
[9] G. Z. and L. H. H. Xu, H. Huang, S. Chen, “Achieving High Scalability Through Hybrid Switching in Software-Defined Networking,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 618–632, 2018, doi: 10.1109/TNET.2018.2789339.
[10] H. Sufiev and Y. Haddad, “A dynamic load balancing architecture for SDN,” *2016 IEEE Int. Conf. Sci. Electr. Eng. ICSEE 2016*, Jan. 2017, doi: 10.1109/ICSEE.2016.7806104.
[11] R. R. Z. Abdulmajeed Adil Yazdeen, Subhi R. M. Zeebaree, Mohammed Mohammed Sadeeq, Shakir Fattah Kak, Omar M. Ahmed, “FPGA Implementations for Data Encryption and Decryption via Concurrent and Parallel Computation: A Review,” *Qubahan Acad. J.*, vol. 1, no. 2, pp. 8–16, 2021, doi: https://doi.org/10.48161/qaj.v1n2a38.
[12] M. A. R. A. Shailendra Mishra, “Software Defined Networking: Research Issues, Challenges and Opportunities,” *Indian J. Sci. Technol.*, vol. 10, no. 29, pp. 1–9, 2017, doi: 10.17485/ijst/2017/v10i29/112447.
[13] Victor Netes, Margarita Kusakina, “Reliability Challenges in Software Defined Networking,” *PROCEEDING 24TH Conf. Fruct Assoc.*, pp. 704–709, 2019, [Online]. Available: https://dl.acm.org/doi/pdf/10.5555/3338290.3338390
[14] D. B. Rawat and S. R. Reddy, “Software Defined Networking Architecture, Security and Energy Efficiency: A Survey,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, pp. 325–346, 2017, [Online]. Available: https://ieeexplore.ieee.org/document/7593247
[15] I. Seremet, S. Hadzovic, S. Mrdovic, and S. Causevic, “SDN as a Tool for Energy Saving,” *27th Telecommun. Forum, TELFOR 2019*, Nov. 2019, doi: 10.1109/TELFOR48224.2019.8971086.
[16] E. Gelenbe and S. Silvestri, “Optimisation of Power Consumption in Wired Packet Networks,” *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng.*, vol. 22 LNICST, pp. 717–729, 2009, doi: 10.1007/978-3-642-10625-5_45.

[17] R. Carpa, O. Gluck, and L. Lefevre, "Segment routing based traffic engineering for energy efficient backbone networks," *2014 IEEE Int. Conf. Adv. Networks Telecommun. Syst. ANTS 2014*, 2014, doi: 10.1109/ANTS.2014.7057272.

[18] Radu Carpa, "Energy Efficient Traffic Engineering in Software Defined Networks," *Netw. Internet Archit. [cs.NI]. Univ. Lyon,* 2017, [Online]. Available: https://hal.science/tel-01650148/

[19] F. A. Moghaddam and P. Grosso, "Linear programming approaches for power savings in software-defined networks," *IEEE NETSOFT 2016 - 2016 IEEE NetSoft Conf. Work. Software-Defined Infrastruct. Networks, Clouds, IoT Serv.*, pp. 83–87, Jun. 2016, doi: 10.1109/NETSOFT.2016.7502448.

[20] O. A. Lagasane Kra, Yaké Gondo, Bi Tra Gooré, "Contribution to the Optimization of the Energy Consumption in SDN Networks," *J. Sens. Technol.*, vol. 8, no. 3, 2018, doi: 10.4236/jst.2018.83005.

[21] D. G. Mehmet Fatih Tuysuz, Zekiye Kubra Ankarali, "A survey on energy efficiency in software defined networks," *Comput. Networks*, vol. 113, pp. 188–204, 2017, doi: https://doi.org/10.1016/j.comnet.2016.12.012.

[22] N. B. Asad Ali Memon, "Performance Analysis of Traditional Networks with Software Defined Networks (SDN) over Various Topologies Using Mininet," *Int. J. Latest Eng. Manag. Res.*, vol. 05, no. 12, pp. 18–26, 2020, [Online]. Available: http://www.ijlemr.com/papers/volume5-issue12/2-IJLEMR-44433.pdf