SDN based OpenFlow Virtualization of 5G Networks

M Vineet Nayak – 231CS132
Prahas G R – 231CS142
Nischal Basavaraju – 231CS139

April 25, 2025

Contents

1 About the Project

The main aim of this project is to demonstrate the working of 5G Network Virtualization using Software Defined Networking (SDN) Controllers. This approach represents a significant evolution from traditional network architecture. In conventional networking systems, the control plane (which makes decisions about where traffic is sent) and the data plane (which forwards traffic to the selected destination) are tightly integrated within the network devices, such as routers and switches. However, in SDN, these two planes are decoupled, which brings flexibility, programmability, and centralized control to the network.

In SDN-enabled networks, the control layer is handled by an SDN controller, which is a centralized software program that manages the flow of data across the network. The forwarding layer is managed by virtual switches such as Open vSwitch (OVS). These virtual switches are not responsible for making routing decisions themselves. Instead, they communicate with the SDN controller via well-defined APIs like OpenFlow, which allows the controller to push flow rules to the switches dynamically.

For this project, we have chosen the following tools to simulate a virtualized 5G environment:

- **Ryu**: An open-source, component-based SDN controller written in Python. It provides a clean interface to interact with OpenFlow-enabled switches.
- **Mininet**: A network emulator that allows rapid prototyping of software-defined networks by simulating virtual hosts, switches, links, and controllers.
- **Open vSwitch**:A multilayer virtual switch designed to enable massive network automation while supporting standard management interfaces and protocols.
- **OpenFlow**: A communications protocol that gives access to the forwarding plane of a switch or router over the network.

Since these tools are designed to run on Linux systems, we utilize Multipass, a lightweight virtual machine, to spin up an Ubuntu VM on macOS.

1.1 SDN Flow Logic

One of the central ideas in SDN is that the switch does not make forwarding decisions. When a packet arrives at a switch and it does not have a corresponding flow rule in its flow table, it sends the packet to the controller. The controller then evaluates the packet's header fields — such as source/destination IP, protocol, or port number — and determines the appropriate action (e.g., forward to a particular port, drop, or modify). It then sends back the action to the switch and installs the flow rule in the switch's flow table for future matching. This process greatly reduces decision-making latency for recurring traffic and enhances performance.

Flow rules enable the controller to dictate how traffic should flow through the network, which is useful for implementing network policies, Quality of Service (QoS), and traffic engineering. Moreover, the SDN controller can also learn and store MAC addresses dynamically, improving efficiency in Layer 2 forwarding.

1.2 Virtualization in 5G Networks

While the SDN controller manages traffic within a single control domain, network virtualization enables the creation of multiple isolated virtual networks on top of a shared physical infrastructure. This is especially critical in 5G networks, where different types of services — such as video streaming, voice calls, IoT connectivity, and mission-critical communications — may coexist but require different performance, security, and control characteristics.

Using SDN, we can assign different flow rules and control logic to different virtual slices of the network. These slices function like Virtual LANs (VLANs), but on a more advanced and scalable level. For instance:

- One slice could handle voice traffic with low latency.
- Another could manage high-bandwidth applications like video streaming.
- Yet another slice might serve IoT devices with lightweight but highly reliable communication needs.

This kind of network slicing is fundamental to 5G architecture, allowing network operators to deliver customized experiences for various use cases — from autonomous vehicles to smart cities.

In our project, we mimic this concept by creating multiple isolated networks within the same Mininet topology, each governed by a common controller but with independent flow rules and communication constraints. These isolated environments do not communicate unless explicitly configured to do so via trunking protocols, enhancing network security and ensuring that traffic from one virtual slice does not interfere with another.

2 Implementation Overview: Simulating 5G Network Virtualization using SDN

To simulate 5G network virtualization using Software Defined Networking (SDN), we have developed and utilized the following key components:

1. **SDN Controller (Ryu-based):** A custom SDN controller is developed using the Ryu framework. This controller is responsible for dynamically managing network flow rules, maintaining flow tables, MAC address tables, and enabling routing across VLANs. It ensures seamless communication between hosts, especially when they are segmented into different virtual networks (VLANs).

2. **Mininet Topology with Virtualization:** In this topology, network virtualization is implemented using VLANs. Hosts are grouped into different VLANs to simulate isolated virtual network slices—an essential component of 5G network architecture. Inter-VLAN communication is enabled through trunk links, and all VLAN-specific routing and flow control are managed by the Ryu SDN controller. This topology effectively demonstrates how network resources can be logically partitioned and centrally managed in a virtualized 5G environment.

3. **Mininet Topology without Virtualization:** This is a traditional flat network topology where all hosts exist on a single broadcast domain without any VLAN-based segmentation. No SDN controller is used in this setup, and routing is handled using default

Mininet behavior or static configurations. This topology serves as a baseline for comparison against the virtualized setup to evaluate the impact and advantages of SDN-based network slicing.

The primary objective of creating two separate topologies is to **evaluate the effectiveness of SDN-driven virtualization** in a simulated 5G environment. The virtualized topology mirrors real-world 5G networks where infrastructure is logically divided for different services or tenants, while the non-virtualized topology represents conventional networks with limited flexibility and programmability.

By conducting various tests and performance measurements (described in the following section), we compare the **efficiency, scalability, and traffic management capabilities** of both architectures. These comparisons help illustrate the benefits of using SDN for dynamic control and orchestration in 5G networks.

3 Results

3.1 Throughput

Observation

To evaluate the throughput of the network, we used the popular network performance measurement tool `iperf`. Tests were conducted on both topologies—virtualized and non-virtualized—by varying the desired bandwidths from 100 Mbps to 10 Gbps.

From the experiments, it was consistently observed that the virtualized network demonstrated significantly better throughput performance than the non-virtualized counterpart. Specifically, the throughput in the virtualized setup remained relatively stable for repeated transmissions of data packets, while in the non-virtualized network, the measured throughput showed large fluctuations for the same test conditions.

In one such instance, for a fixed data packet size, the throughput in the non-virtualized network ranged from as low as 515 Mbits/sec to as high as 4.18 Gbits/sec. This inconsistency was not present in the virtualized network, which maintained a steady and predictable throughput under identical conditions.

Reason

The improved and stable throughput performance in the virtualized network can be attributed to the architectural advantages of SDN and Open vSwitch (OVS). In SDN-enabled networks, once flow entries are installed into the switch by the controller, packet forwarding occurs in the kernel-level fast path of OVS. This means that subsequent packets of a flow bypass the user space and are handled entirely in kernel space, drastically reducing the overhead caused by user-kernel context switching.

Furthermore, SDN allows for intelligent, dynamic control over routing and link utilization. This means that traffic can be distributed more efficiently across the network, reducing congestion and ensuring better load balancing. These mechanisms directly contribute to minimizing packet processing delays and increasing the effective throughput.

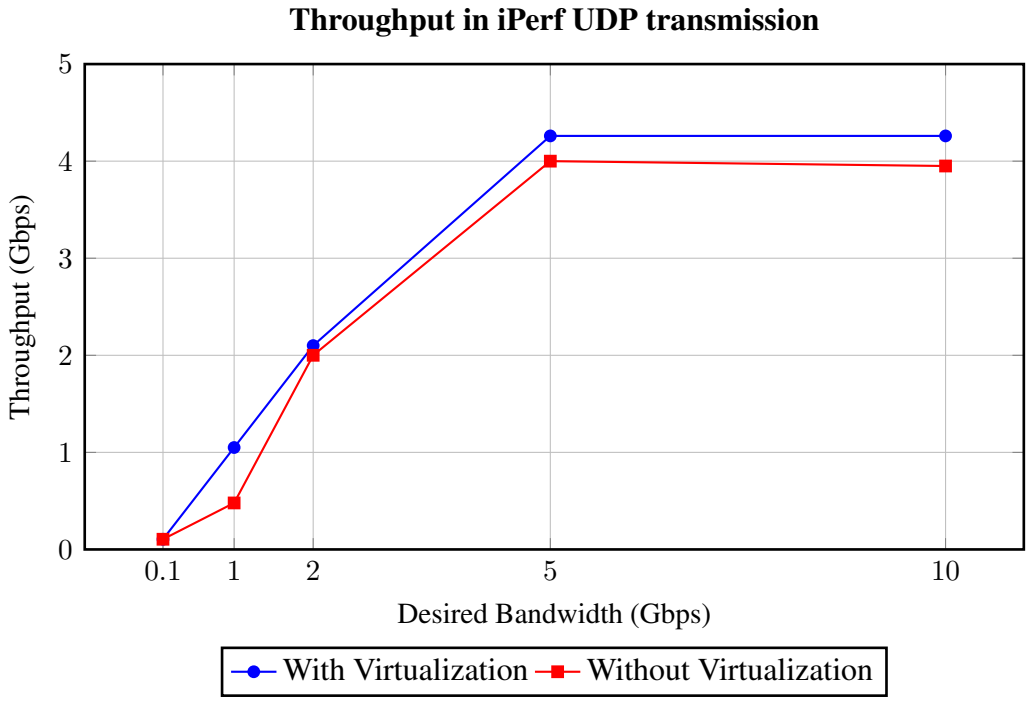
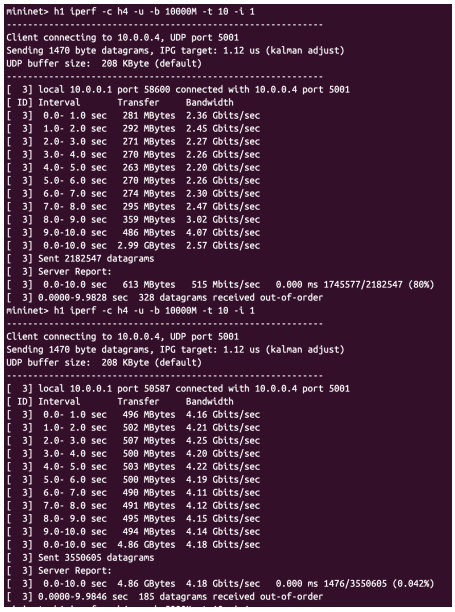


Figure 1: Throughput comparison of virtualized vs non-virtualized network environments



```
mininet> h1 iperf -c h4 -u -b 10000M -t 10 -l 1
------------------------------------------------------------
Client connecting to 10.0.0.4, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1.12 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58600 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec   281 MBytes  2.36 Gbits/sec
[  3]  1.0- 2.0 sec   292 MBytes  2.45 Gbits/sec
[  3]  2.0- 3.0 sec   271 MBytes  2.27 Gbits/sec
[  3]  3.0- 4.0 sec   270 MBytes  2.26 Gbits/sec
[  3]  4.0- 5.0 sec   263 MBytes  2.20 Gbits/sec
[  3]  5.0- 6.0 sec   270 MBytes  2.26 Gbits/sec
[  3]  6.0- 7.0 sec   274 MBytes  2.30 Gbits/sec
[  3]  7.0- 8.0 sec   295 MBytes  2.47 Gbits/sec
[  3]  8.0- 9.0 sec   359 MBytes  3.02 Gbits/sec
[  3]  9.0-10.0 sec   486 MBytes  4.07 Gbits/sec
[  3]  0.0-10.0 sec  2.99 GBytes  2.57 Gbits/sec
[  3] Sent 2182547 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec   613 MBytes   515 Mbits/sec   0.000 ms 1745577/2182547 (80%)
[  3] 0.0000-9.9828 sec  328 datagrams received out-of-order
mininet> h1 iperf -c h4 -u -b 10000M -t 10 -l 1
------------------------------------------------------------
Client connecting to 10.0.0.4, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1.12 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 50587 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec   496 MBytes  4.16 Gbits/sec
[  3]  1.0- 2.0 sec   502 MBytes  4.21 Gbits/sec
[  3]  2.0- 3.0 sec   507 MBytes  4.25 Gbits/sec
[  3]  3.0- 4.0 sec   500 MBytes  4.20 Gbits/sec
[  3]  4.0- 5.0 sec   503 MBytes  4.22 Gbits/sec
[  3]  5.0- 6.0 sec   500 MBytes  4.19 Gbits/sec
[  3]  6.0- 7.0 sec   490 MBytes  4.11 Gbits/sec
[  3]  7.0- 8.0 sec   491 MBytes  4.12 Gbits/sec
[  3]  8.0- 9.0 sec   495 MBytes  4.15 Gbits/sec
[  3]  9.0-10.0 sec   494 MBytes  4.14 Gbits/sec
[  3]  0.0-10.0 sec  4.86 GBytes  4.18 Gbits/sec
[  3] Sent 3550605 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec  4.86 GBytes  4.18 Gbits/sec   0.000 ms 1476/3550605 (0.042%)
[  3] 0.0000-9.9846 sec  185 datagrams received out-of-order
```

Figure 2: Variation observed in Throughput for non-virtualized networks.

The figure above illustrates the throughput variations captured during testing. The virtualized network clearly demonstrates superior consistency and overall performance across varying bandwidth levels. This highlights the benefits of virtualization and SDN control in managing high-throughput, low-latency network environments—an essential requirement for 5G applications.

3.2 Data Packet Loss

Observation

Virtualized networks consistently outperform non-virtualized networks when it comes to managing data packet loss, particularly under varying network loads.

The performance differences become especially evident when we observe the data packet loss metrics. As illustrated in the previous figure, one test run in the non-virtualized network showed a significant packet loss of 80%, while the virtualized network exhibited a negligible loss of 0.042%. This stark contrast highlights the superior handling of network traffic in virtualized environments.

Subsequent tests repeated the same conditions, and the data packet loss in the virtualized network remained largely stable, fluctuating only between 0.3% and 0.5%. These fluctuations, while noticeable, were minor compared to the dramatic packet loss observed in the non-virtualized network. More importantly, no severe or drastic increases in packet loss were recorded in the virtualized setup, demonstrating its robustness under load. In contrast, the non-virtualized network continued to show erratic spikes in packet loss, especially under heavier traffic conditions.

Reason

The primary reason behind the low and stable packet loss in the virtualized network lies in the combination of Open vSwitch (OVS) and SDN control. OVS, when managed by an SDN controller, provides several advantages that help in reducing packet loss:

1. **Per-flow Buffering:** OVS is capable of managing traffic on a per-flow basis, meaning each individual flow of data is buffered and processed separately. This allows for more precise management of traffic patterns, which helps prevent congestion and packet drops.

2. **Quality of Service (QoS) Policies:** With SDN, QoS policies can be implemented dynamically and efficiently. These policies smooth out traffic bursts and ensure that the network can handle peaks in traffic without overloading buffers. This is a crucial feature that reduces packet loss in real-time by prioritizing more critical or time-sensitive data flows.

3. **Congestion Avoidance:** The SDN controller has a global view of the network, meaning it can observe traffic patterns across the entire system. This allows the controller to anticipate potential congestion issues before they occur, such as when a queue is about to overflow. It can reroute traffic dynamically, ensuring that traffic flows smoothly without dropping packets. In contrast, traditional non-virtualized networks lack such dynamic intelligence and typically rely on static queuing mechanisms that cannot respond effectively to sudden bursts in traffic.

4. **Load Balancing and Traffic Management:** SDN enables advanced traffic management techniques such as load balancing, where traffic is distributed across multiple paths to avoid congestion. This further reduces the likelihood of packet loss, especially in networks with high traffic volumes.

In a non-virtualized setup, there is typically no central management system capable of handling individual traffic flows with this level of granularity. These networks rely on static, fixed

routing and queuing techniques that are often unable to cope with the complexities of modern data traffic, leading to higher and more unpredictable packet loss.

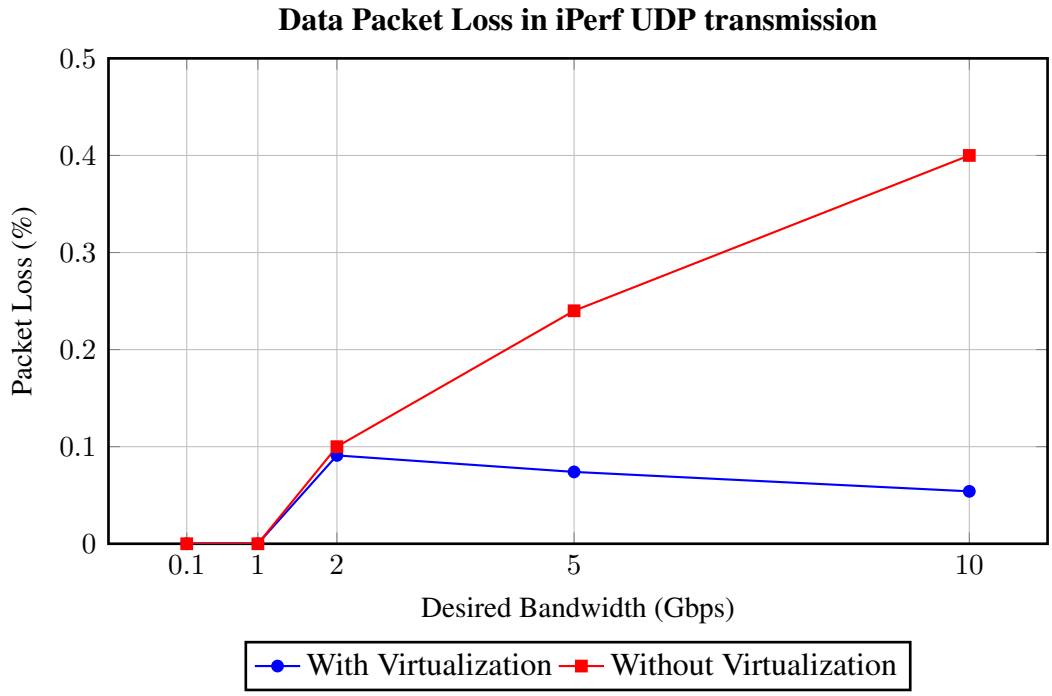


Figure 3: Data Packet Loss comparison of virtualized vs non-virtualized network environments

In summary, the ability of the virtualized network, controlled by SDN and enhanced with OVS, to provide dynamic, fine-grained traffic management results in significantly lower and more predictable data packet loss. This capability is essential for modern, high-performance networks, particularly in environments requiring low-latency communication and high reliability, such as 5G networks.

3.3 Out-of-Order Datagrams

Observation

In our testing, a significant difference in packet sequencing was observed between virtualized and non-virtualized networks. In the virtualized network, there were no out-of-order data frames, meaning all packets arrived in the same order they were sent. However, in the non-virtualized network, a noticeable number of out-of-order data frames were recorded, indicating that packets were being delivered in a sequence different from their intended order.

This behavior in the non-virtualized network can lead to delays in data processing and can complicate applications that rely on strict sequencing, such as real-time communication or streaming services. The issue of out-of-order frames is less prevalent in virtualized environments, where packet delivery is more predictable and reliable.

Reason

The reason for this stark contrast lies in the control mechanisms provided by Software Defined Networking (SDN) and the deterministic nature of Open vSwitch (OVS) in virtualized

environments. Several factors contribute to the absence of out-of-order frames in virtualized networks:

1. **Centralized Control by SDN Controllers:** One of the key features of SDN is the ability to have a centralized controller that manages the entire network's routing decisions. This controller can enforce consistent forwarding rules, ensuring that all packets belonging to the same flow traverse the same path through the network. With SDN, there are no last-minute changes in routing decisions because the paths are predetermined and optimized for the flow of data. This results in packets being delivered in the exact order they were transmitted, as there is no ambiguity in the route they take.

2. **Deterministic Queueing with OVS:** Open vSwitch (OVS) provides deterministic queueing in the kernel datapath. This means that each flow of traffic is handled in a predictable manner without random reordering. By having a single, optimized path for each flow, OVS eliminates path variability and ensures that all packets follow the same queueing schedule. When packets are processed in the kernel-level fast path, the kernel manages them in an ordered manner, preventing the possibility of out-of-order frame delivery.

3. **Lack of Dynamic Path Changes:** In traditional, non-virtualized networks, packets may be forwarded through multiple paths based on available network conditions, routing updates, or congestion avoidance techniques. Without the centralized intelligence of an SDN controller, the forwarding paths can change unexpectedly during the transmission of packets. These last-minute route changes, often due to fluctuating network conditions or dynamic load balancing, introduce the potential for packets to arrive at their destination in a different order than they were sent.

4. **Scheduling Jitter:** In non-virtualized networks, the forwarding and scheduling of packets are typically handled by the switch or router's local algorithms, which may lack the precision needed to maintain order. As a result, scheduling jitter can occur, where packets are queued and transmitted at uneven intervals, leading to variations in arrival times. This can result in packets arriving out of sequence, further exacerbating the issue of out-of-order frames.

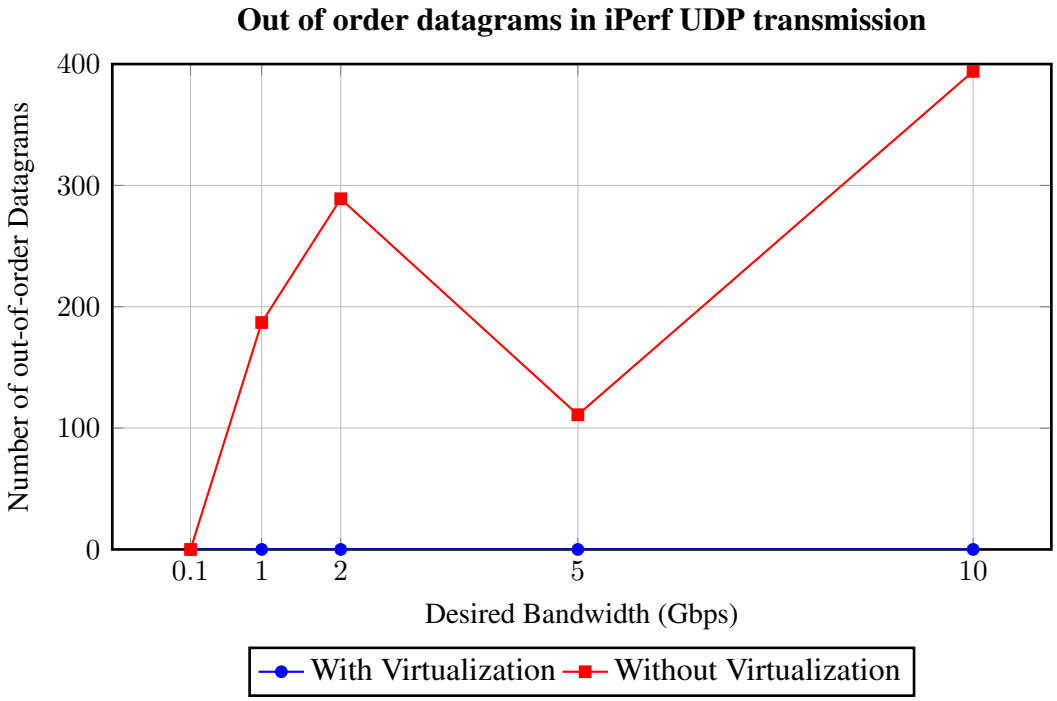


Figure 4: Out of order datagrams comparison of virtualized vs non-virtualized network environments

In summary, the absence of out-of-order data frames in virtualized networks can be attributed to the deterministic and consistent behavior of SDN-controlled networks. The centralized control of SDN ensures that traffic flows are managed optimally and predictably, while OVS guarantees that packets follow the same, reliable path. In contrast, non-virtualized networks, lacking such centralized control, are prone to unpredictability in packet routing and scheduling, leading to increased occurrences of out-of-order frame delivery. This discrepancy highlights one of the key advantages of SDN and network virtualization in providing reliable, high-performance communication.

4 Conclusion

Through extensive testing and configuration of different network setups, it is clear that network virtualization using Software Defined Networking (SDN) offers numerous advantages over traditional, non-virtualized networks. These benefits span across performance, scalability, ease of maintenance, and flexibility, making SDN a compelling choice for modern networking, particularly for high-performance environments like **5G**.

One of the primary advantages of virtualized networks is their **modularity**. Traditional networks, with their rigid architecture, can be difficult to modify or expand without significant changes to the physical infrastructure. In contrast, SDN-enabled virtualized networks are far more modular, allowing for easier adjustments to accommodate new technologies, services, and network demands. This modularity significantly reduces the complexity of maintaining and upgrading networks, making them more adaptable to evolving requirements. This flexibility is especially beneficial in the context of rapidly changing fields like 5G, where agility and scalability are key.

In terms of **performance**, our tests have shown that virtualized networks consistently outperform their non-virtualized counterparts. The **throughput** in virtualized networks was no-

tably more stable and higher, even under heavy load, compared to the non-virtualized network. This is a direct result of the dynamic traffic management and optimized path selection enabled by SDN. The ability of SDN to monitor and control the network in real-time allows for better congestion management and load balancing, ensuring that data is transmitted efficiently and reliably. Moreover, the lack of significant **packet loss** in virtualized networks further enhances their suitability for performance-sensitive applications. Unlike traditional networks, where packet loss can be substantial due to static routing and insufficient traffic management, SDN-controlled networks handle data more efficiently, minimizing data loss even under high congestion.

Furthermore, **out-of-order packet delivery**—which is common in non-virtualized networks due to path variability and scheduling jitter—was absent in the virtualized network. This guarantees that data is transmitted in the exact sequence it was sent, which is critical for applications that require high reliability and consistent data delivery, such as real-time communication, video streaming, and other latency-sensitive services. The deterministic and consistent packet flow offered by SDN and **Open vSwitch (OVS)** is an essential feature that contributes to the superior performance and reliability of virtualized networks.

The **centralized control** provided by SDN not only ensures consistent packet delivery but also allows for the implementation of sophisticated **Quality of Service (QoS)** policies. These policies enable better traffic management, preventing congestion and ensuring that high-priority traffic is given the necessary bandwidth, further improving overall network performance.

In addition to performance improvements, **maintenance** of SDN-based virtualized networks is significantly easier compared to traditional setups. Traditional networks often require manual intervention and configuration of each device in the network. In contrast, SDN allows for centralized management, where changes can be made to the entire network through software updates. This centralized approach not only reduces the risk of human error but also simplifies network troubleshooting and scaling. As network demands grow, SDN provides the tools to reconfigure and expand the network with minimal disruption, making it a future-proof solution for rapidly expanding technologies like 5G.

Finally, the modularity, scalability, and reliability of SDN-based virtualized networks make them well-suited for **5G** applications, which demand **low latency**, **high throughput**, and **high reliability**. The ability to dynamically allocate resources, optimize routing paths, and manage traffic in real time is essential for supporting the ultra-low latency and high-speed requirements of 5G networks. By leveraging SDN, network operators can ensure that 5G services meet the stringent performance demands of consumers and businesses alike.

In conclusion, the results of our tests and configurations clearly demonstrate that SDN and network virtualization provide significant advantages over traditional networks. These advantages make virtualized networks not only more efficient and stable but also more adaptable, ensuring they are better suited for the next generation of high-performance applications like 5G. The combination of **modularity**, **improved performance metrics**, **lower packet loss**, and **easier maintenance** positions SDN as the optimal solution for future networking challenges.

These characteristics make SDN-based virtualization ideal for high-performance applications like **5G networks**.

5 Supporting Documents

For the complete project code, configurations, and additional details, please click here or visit: https://docs.google.com/document/d/1RJ7bdljxD6K7EEaGClzsRmZ9W1ccXww5Aindq_hSUXM/edit?usp=sharing

The following resources are included:

- **Ryu Controller Code:** A comprehensive implementation of the SDN controller using Ryu for managing network flows and ensuring optimal packet routing.
- **Mininet Topology with Virtualization:** The configuration of a Mininet topology incorporating SDN and network virtualization using VLANs, including trunking for inter-VLAN communication.
- **Mininet Topology without Virtualization:** A basic Mininet setup for performance comparison, focusing on non-virtualized network architecture.

A research paper [1] was used for basic understanding of Software Defined Networks. The tests carried out here are not part of this paper.

References

[1] Abdelmoneim A Bakhit, Sharifah HS Ariffin, Mutaz HH Khairi, Siti Zarina, Abu Ubaidah Shamsuddin Muji, and Mohd Fadzli Sahib. Network implementation based on software defined network using ryu controller and openvswitch.