7COM1076-0901-2022

Wireless Mobile and Multimedia Networking

Coursework -01

Table of Contents

1. Introduction:

Mininet

In a software-defined network, Mininet is one of the most widely used tools based on this network technology. This device acts as an emulator which can be used like that an emulator. One can use this to visualize various switches which is one of the foremost functions of a software-defined network. This is tool is used worldwide to make various performances for different experiments one must perform (Hong *et al.,* 2020). This is done to give a better understanding of the learning tool that is used to test furthermore helping the user to learn more about software-defined networks. Mininet helps the user to get a faster and more customized technological service that helps them to maintain the OpenFlow switches. Hence this service tool has won the preference of most the users when compared to other services available. It installs a virtual machine along with a virtual box that can perform all the functions required to be performed. With Mininet, the internal network functions are handled with ease.

ONOS

ONOS stands for open network operating system. This is identified as one of the most efficient open Source SDN controllers that have become a necessity to build the next generation SDV/NFV solutions making it the leading open source worldwide. It has been designed in a way to meet the needs of operators that aim at building career-grade solutions. ONOS can eliminate the requirement of run routing and switching control protocols which in turn saves a lot of time for the user. This can be done as ONOS can support both configuration and real-time control of the given network (Monika *et al.,* 2020). It has been concluded that in ONOS cloud controller,

the movement and innovation have become a subject of networking operation for the end users facility. ONOS is one of the most reliable sources when it comes to the operation of such subjects.

Mininet-wifi

When it comes to managing various network functions, Mininet-wifi can be considered to be the most efficient network in a mininet SDN network emulator. Mininet-wifi is relied upon widely when it comes to visualizing switches and application of software-defined networks. Here, the OpenFlow protocols present within the wifi is used as a network device that is self-defined thus maintaining the flow of operation. With the help of a Hybrid Physical Virtual and software Defined wireless networking strategy, the user can develop a wireless driver improving the operations of a wifi station management.

2. Practical Coursework

a. Task1:

i. Methodology

Task: 1

1.

Name	MAC	IP Address	(X, Y) coordinates	SSID	Password	Model	Range	Channel
AP1	00:00:10:10:20:10	N/A	(60,60,0)	ap1	21079100	DI524	30	1

AP2	00:00:10:10:20:11	N/A	(120,60,0)	ap2	21079100	DI524	30	2
AP3	00:00:10:10:20:12	N/A	(150,60,0)	ap3	21079100	DI524	30	3
AP4	00:00:10:10:20:13	N/A	(160,100,0)	ap4	21079100	DI524	30	6
AP5	00:00:10:10:20:14	N/A	(160,150,0)	ap5	21079100	DI524	30	4
STA1	00:00:10:18:00:02	192.168.110.13/24	(30,50,0)	N/A	21079100	N/A	25	N/A
STA2	00:00:10:18:00:03	192.168.110.14/24	(30,50,0)	N/A	21079100	N/A	25	N/A
STA3	00:00:10:18:00:04	192.168.110.15/24	(30,50,0)	N/A	21079100	N/A	25	N/A
STA4	00:00:10:18:00:05	192.168.110.16/24	(30,50,0)	N/A	21079100	N/A	25	N/A

Name	Start Location	End Location	Start Time-End Time	Moving Speed (min-max)
STA1	(30,50,0)	(140, 80, 0)	10s-20s	min_v=1, max_v=5
STA2	(30,50,0)	(100,50,0)	30s-60s	min_v=5, max_v=10
STA3	(30,50,0)	(170,75, 0)	25s-60s	min_v=2, max_v=7
STA4	(30,50,0)	(50,50,0)	10s-20s	min_v=2, max_v=7

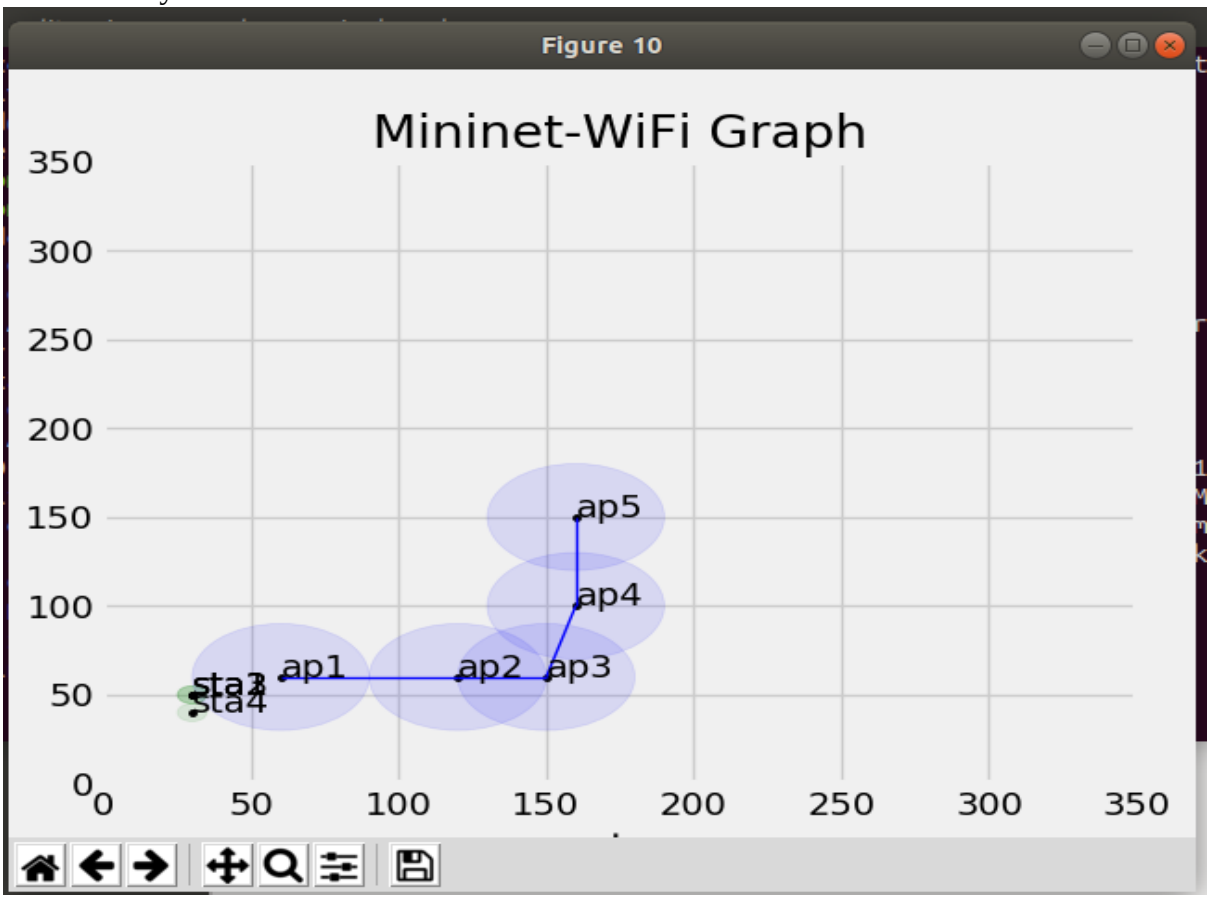
On a closer look at task 1, there is a requirement of 5 wifi access points as shown in the floor plan of a new building which could be accessed by the users in concern. These 5 access points are needed to emulate the environment for which one needs to write the function in a python script. These 5 wifi access points can be installed efficiently using the Mininet tool (Gupta *et al.*, 2022). The UDPS network followed by the previously installed access points can be executed efficiently including all the network functions. To work with the server, the given configuration must work by developing mininet tools, access points, and service stations. The user in concern can develop a python script to initiate the deployment process (DeCusatis, Carranza & Delgado-Caceres, 2016)

ii. Experiments

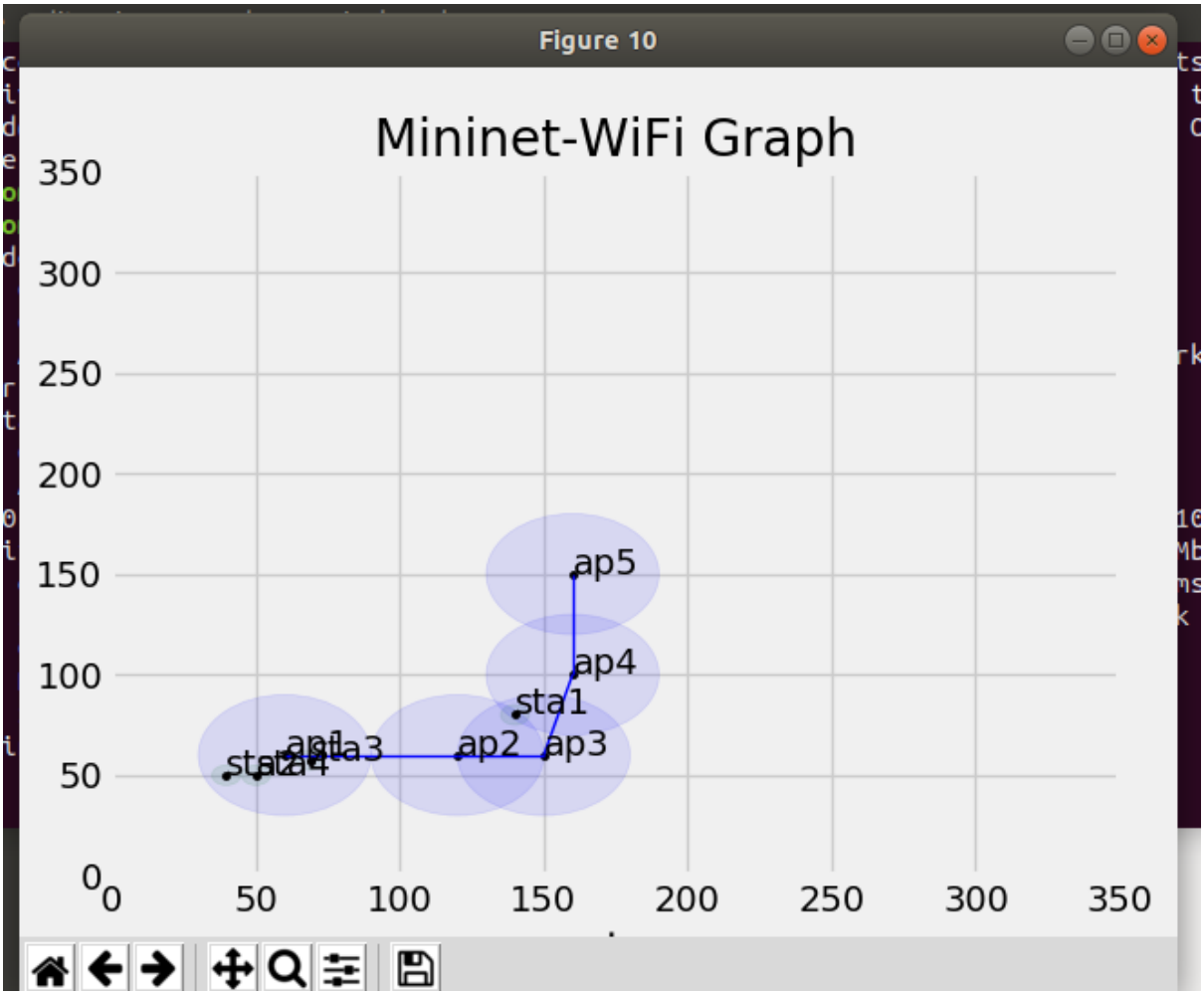
Observing the illustrated plan, one can note the location of the five major access points that are to be used for emulating purposes. It is also observed that these emulation purposes are solely

emulating 4 stations. Stations STA1, STA2, STA3, and STA4 are required to be masked as a Class C IP address. The APS that is required in this plan has to be connected with the use of physical links facilitating the linear topology (Pang *et al.*, 2020). To make sure that the operation is precisely fulfilled as well as attention is clear, a UDP connection must be present between STA2 and UDPS while the UDP server takes the job to connect the network between different networking devices. Below are the screenshots of Mininet Wi-Fi GUI:

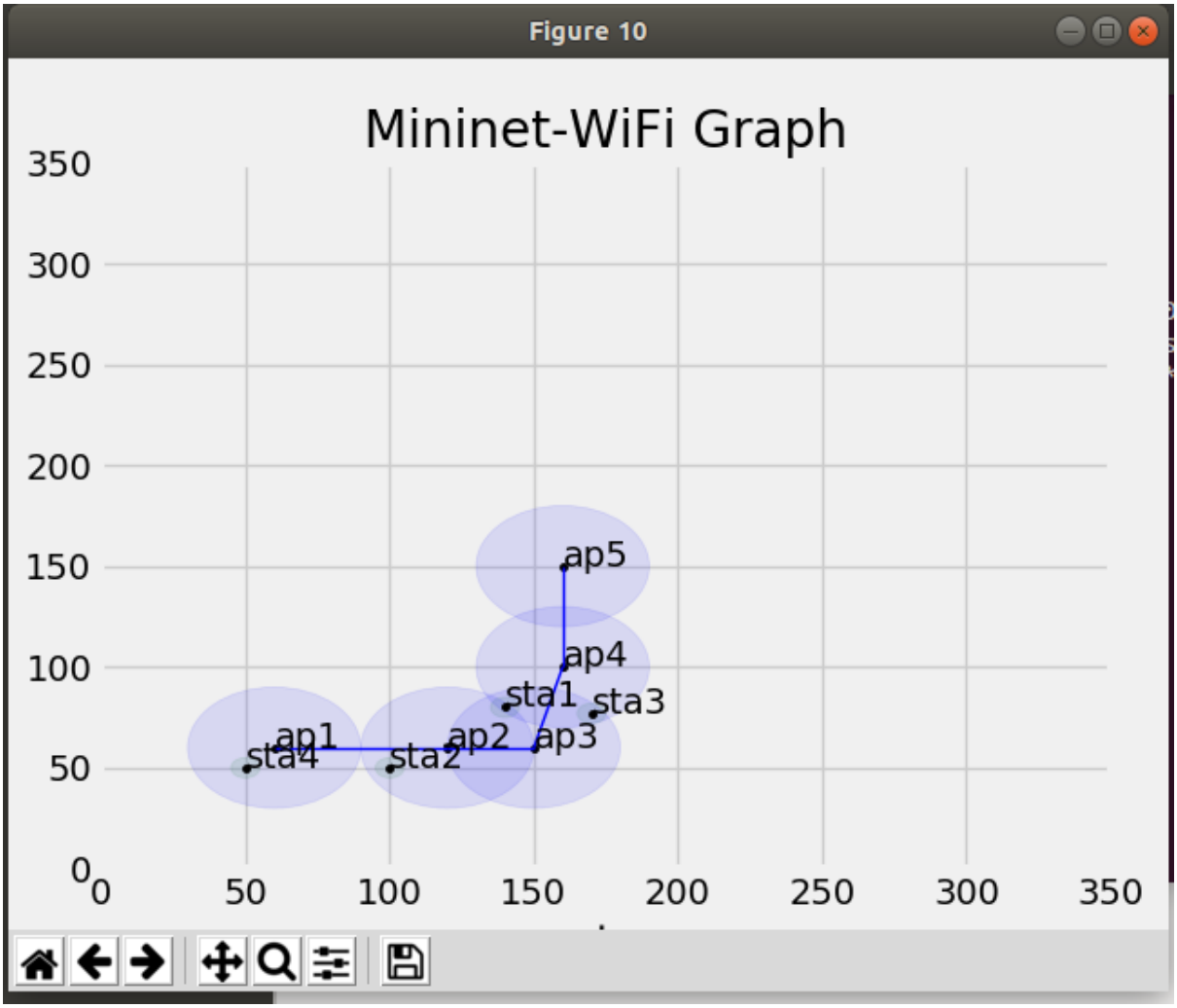
- Prior Mobility:



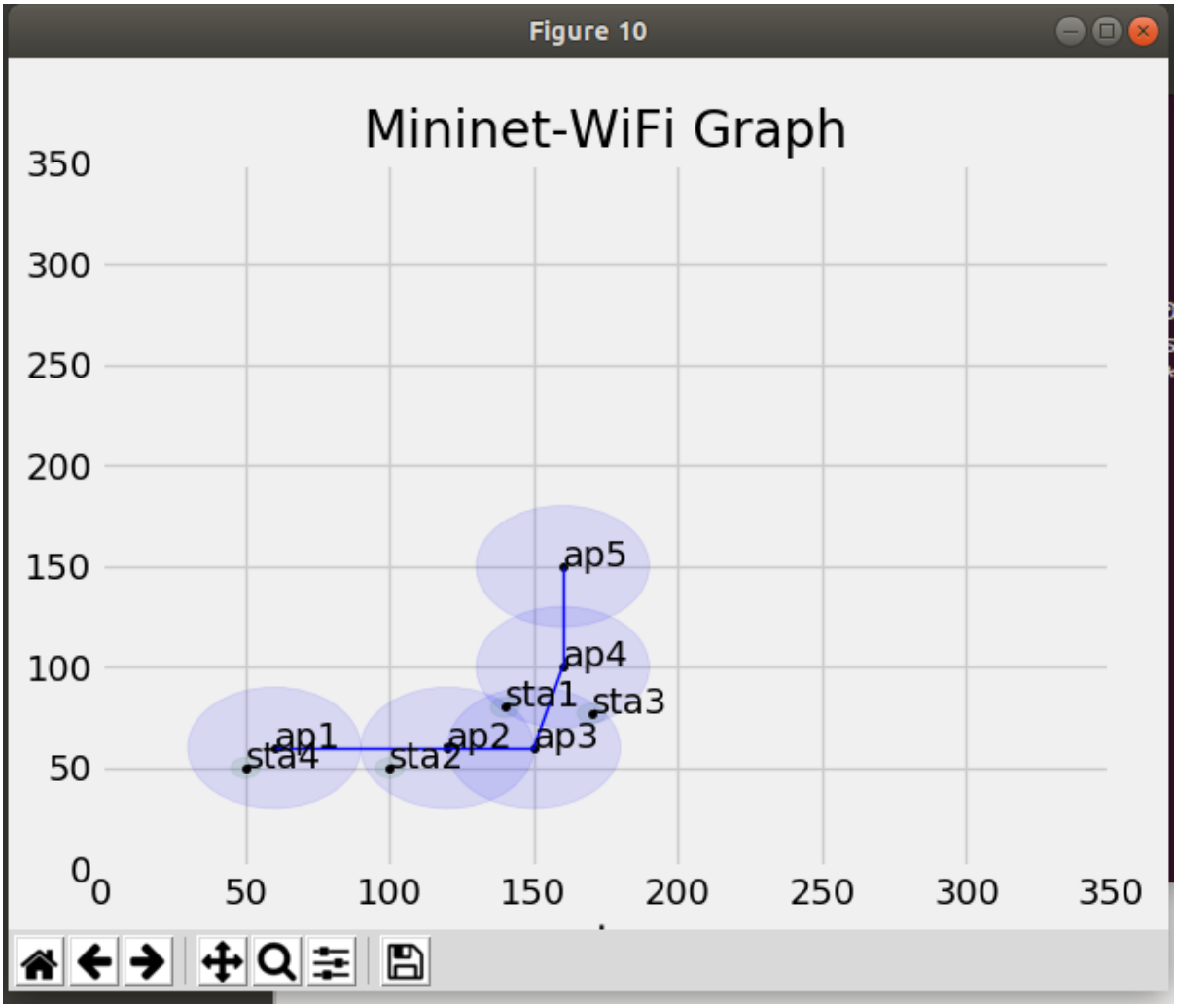
- During Mobility:



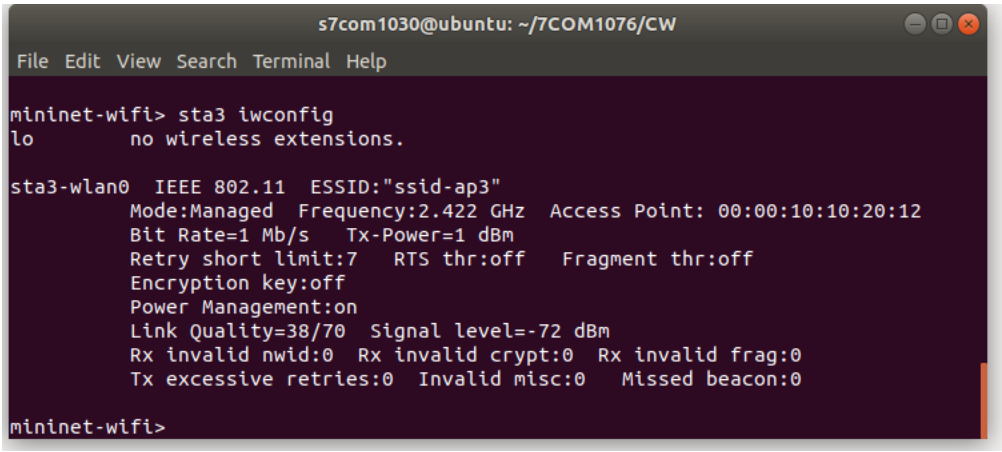
- At the completion of the mobility:



3. Below is the screenshot of connection of access point after the completion of the mobility by stations:



Below is the screenshot of the access point connected upon completion of mobility by stations:



The command that used to obtain that information is:- sta3 iwconfig .

4.

Below is the screenshot of successful ping of Sta1 from Sta2:

```
mininet-wifi> sta1 ping sta2
PING 192.168.110.14 (192.168.110.14) 56(84) bytes of data.
64 bytes from 192.168.110.14: icmp_seq=1 ttl=64 time=15.9 ms
64 bytes from 192.168.110.14: icmp_seq=2 ttl=64 time=6.07 ms
64 bytes from 192.168.110.14: icmp_seq=3 ttl=64 time=6.52 ms
^C
--- 192.168.110.14 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 6.079/9.514/15.940/4.548 ms
```

- Below is the screenshot of successful ping of Sta2 from Sta3:

```
mininet-wifi> sta2 ping sta3
PING 192.168.110.15 (192.168.110.15) 56(84) bytes of data.
64 bytes from 192.168.110.15: icmp_seq=1 ttl=64 time=85.1 ms
64 bytes from 192.168.110.15: icmp_seq=2 ttl=64 time=17.0 ms
^C
--- 192.168.110.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 17.009/51.085/85.161/34.076 ms
```

- Below is the screenshot of successful ping of Sta1 from Sta3:

```
mininet-wifi> sta1 ping sta3
PING 192.168.110.15 (192.168.110.15) 56(84) bytes of data.
64 bytes from 192.168.110.15: icmp_seq=1 ttl=64 time=131 ms
64 bytes from 192.168.110.15: icmp_seq=2 ttl=64 time=11.4 ms
^C
--- 192.168.110.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 11.448/71.672/131.896/60.224 ms
```

- Below is the screenshot of successful ping of Sta1 from Sta4:

```
mininet-wifi> sta1 ping sta4
PING 192.168.110.16 (192.168.110.16) 56(84) bytes of data.
64 bytes from 192.168.110.16: icmp_seq=1 ttl=64 time=67.6 ms
64 bytes from 192.168.110.16: icmp_seq=2 ttl=64 time=12.0 ms
^C
--- 192.168.110.16 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 12.025/39.855/67.686/27.831 ms
```

iii. Evidence and Screenshots

According to task 1, the code is designed, and a snippet has been given for the same.

- The snapshot of the given code explains the significance of python software that has been applied abundantly for task 1 that helps to manage the wifi compatible device. In the given network architecture, there use of various access points and networking for functional applications.
- It is mandatory to have a station in a mobility state.
- For communication, project encryption is the major motivator.
- Wpa2 and the fail Mode are standalone (Muqaddas et al. 2016)
- The average jitter calculated is 65.45 milliseconds.

iv. Analysis and discussion

One can handle the amount of data that can be transmitted from the system based on the networking structure. The packet loss is inversely proportional to bandwidth reduction. Thus, it can be said that packet loss is the result of jitter effects in an overall system implementation.

b. Task 2:

i. Methodology

In this assignment, one can understand from the floor plan of the building and emergency gathering car parking, it is a requirement of a minimum of 3 device mobility. The operator requires 4 Adhoc stations as well as a python script file is required for this to modify the required operation to what the plan desires (Hedges *et al.,* 2019). Initiation of ICMP steam is lying between the sta4ad < - - -> sta5ad, sta5ad < - - -> sta6ad and sta4ad < - - -> sta6ad.

ii. Experiments

It has been proved that maintaining the connection that determines the configuration towards The Mininet Wifi python script is highly regarded and utilized. In the case of ICMP maintenance, one must use a stream to establish the connection between sta5adhoc and sta6adhoc. One must also perform managing experiments in UDP transmission which will help the user to understand jitter and packet loss.

iii. Evidence and Screenshots

The network design of task 2 is demonstrated in the following screenshot.

1.

Name	IPv6	MAC	Position	Range	Antenna Height	Antenna Gain	Protocol	SSID	HT_CAP
sta4ad	fe65::21	00:00:10:20:15:06	175,175,0	40	1	5	olsr	adhocUH	HT40+
sta5ad	fe65::22	00:00:10:20:15:07	195,175,0	40	2	6	olsr	adhocsUH	HT40+
sta6ad	fe65::23	00:00:10:20:15:08	225,175,0	40	3	7	olsr	adhocUH	HT40+
sta7ad	fe65::24	00:00:10:20:15:09	55,175,0	30	3	7	babel	adhocUOH	HT40+
sta8ad	fe65::25	00:00:10:20:15:10	75,175,0	30	3	7	babel	adhocUOH	HT40+
sta9ad	fe65::26	00:00:10:20:15:11	100,175,0	30	3	7	babel	adhocUOH	HT40+

2.

- Below is the screenshot of successful ICMP Stream between sta4ad and sta5ad:

```
mininet-wifi> sta4adhoc ping sta5adhoc
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=3 ttl=64 time=215 ms
64 bytes from 10.0.0.6: icmp_seq=6 ttl=64 time=43.9 ms
64 bytes from 10.0.0.6: icmp_seq=7 ttl=64 time=2.13 ms
^C
--- 10.0.0.6 ping statistics ---
7 packets transmitted, 3 received, 57% packet loss, time 6091ms
rtt min/avg/max/mdev = 2.130/87.320/215.929/92.525 ms
```

- Below is the screenshot of successful ICMP Stream between sta5ad and sta6ad:

```
mininet-wifi> sta5adhoc ping sta6adhoc
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=216 ms
64 bytes from 10.0.0.7: icmp_seq=6 ttl=64 time=107 ms
64 bytes from 10.0.0.7: icmp_seq=7 ttl=64 time=10.4 ms
^C
--- 10.0.0.7 ping statistics ---
7 packets transmitted, 3 received, 57% packet loss, time 6079ms
rtt min/avg/max/mdev = 10.486/111.416/216.567/84.185 ms
```

- Below is the screenshot of successful ICMP Stream between sta4ad and sta6ad:

```
mininet-wifi> sta4adhoc ping sta6adhoc
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=215 ms
64 bytes from 10.0.0.7: icmp_seq=6 ttl=64 time=108 ms
64 bytes from 10.0.0.7: icmp_seq=7 ttl=64 time=2.05 ms
^C
--- 10.0.0.7 ping statistics ---
7 packets transmitted, 3 received, 57% packet loss, time 6090ms
rtt min/avg/max/mdev = 2.059/108.565/215.620/87.186 ms
```

3.

- Below is the screenshot of successful ICMP Stream between sta7ad and sta8ad:

```
mininet-wifi> sta7adhoc ping sta8adhoc
bash: babeld: command not found
PING 10.0.0.9 (10.0.0.9) 56(84) bytes of data.
64 bytes from 10.0.0.9: icmp_seq=3 ttl=64 time=219 ms
64 bytes from 10.0.0.9: icmp_seq=6 ttl=64 time=107 ms
^C
--- 10.0.0.9 ping statistics ---
6 packets transmitted, 2 received, 66% packet loss, time 5082ms
rtt min/avg/max/mdev = 107.619/163.797/219.975/56.178 ms
```

- Below is the screenshot of successful ICMP Stream between sta8ad and sta9ad:

```
mininet-wifi> sta8adhoc ping sta9adhoc
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=220 ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=64 time=75.8 ms
^C
--- 10.0.0.10 ping statistics ---
6 packets transmitted, 2 received, 66% packet loss, time 5089ms
rtt min/avg/max/mdev = 75.875/148.063/220.251/72.188 ms
```

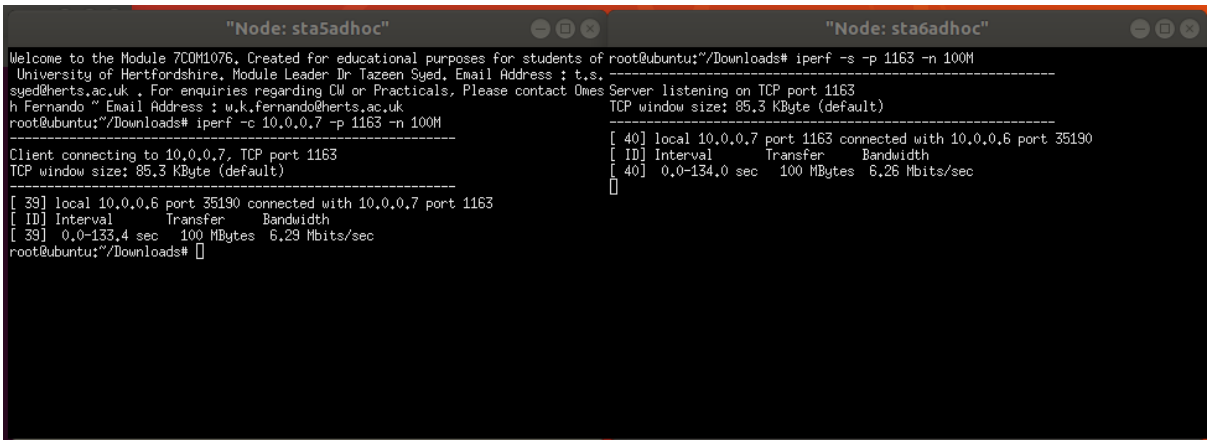
- Below is the screenshot of successful ICMP Stream between sta7ad and sta9ad:

```
mininet-wifi> sta7adhoc ping sta9adhoc
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=215 ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=64 time=108 ms
64 bytes from 10.0.0.10: icmp_seq=7 ttl=64 time=1.44 ms
^C
```

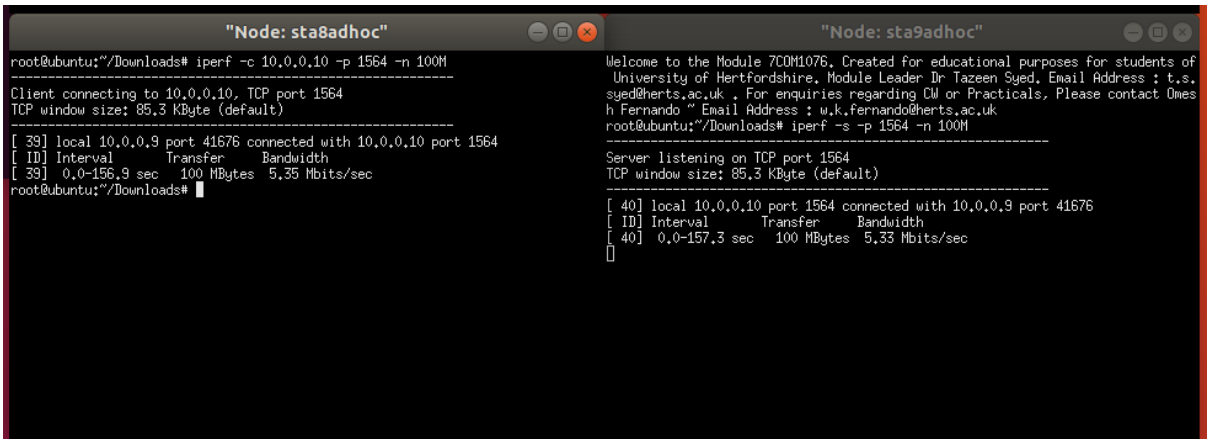
4.

a.

- Below is the screenshot of TCP transfer among server- sta6adhoc and client – sta5adhoc on port 1163:

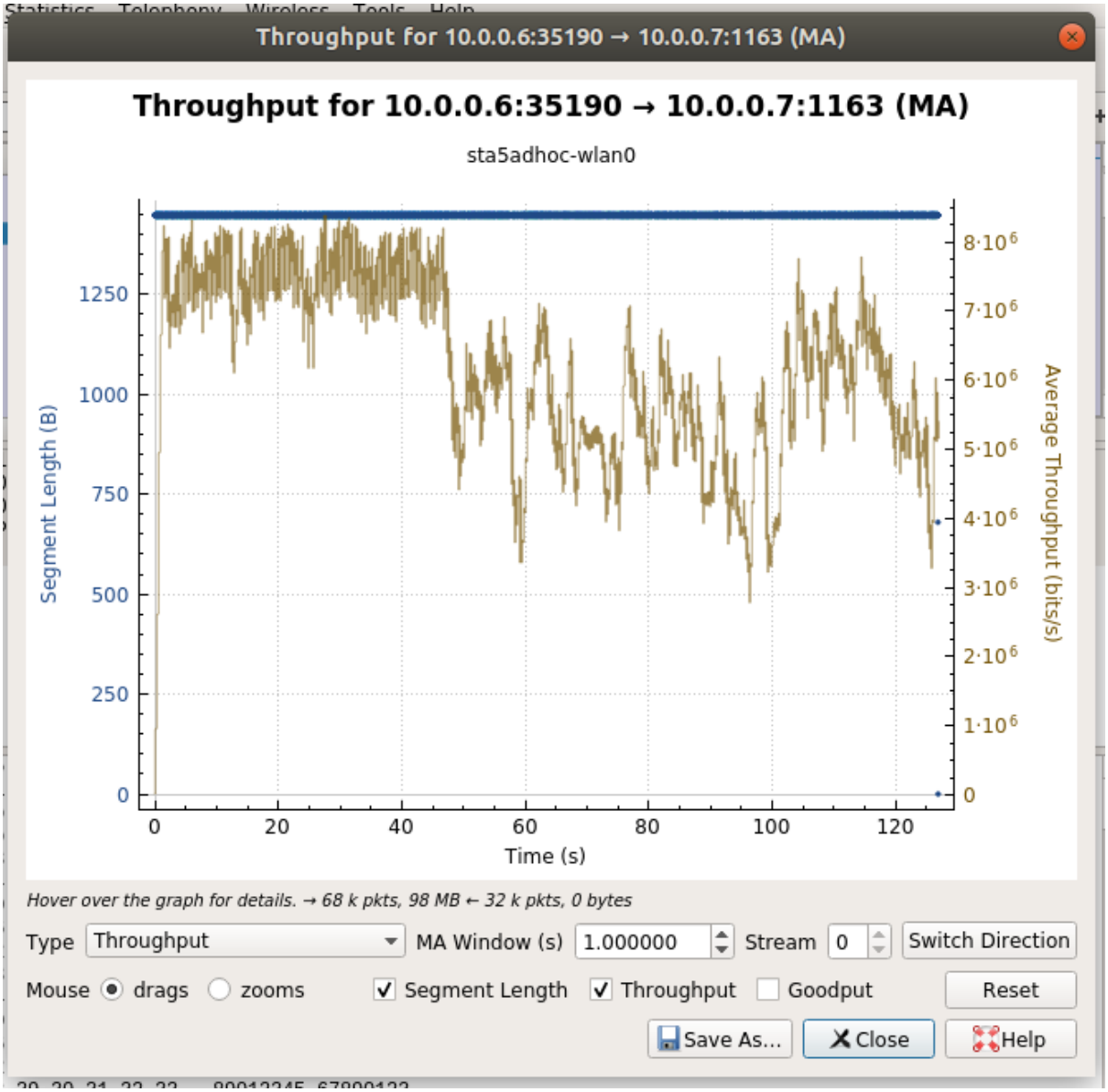


- Below is the screenshot of TCP transfer among server- sta9adhoc and client – sta8adhoc on port 1564:



b.

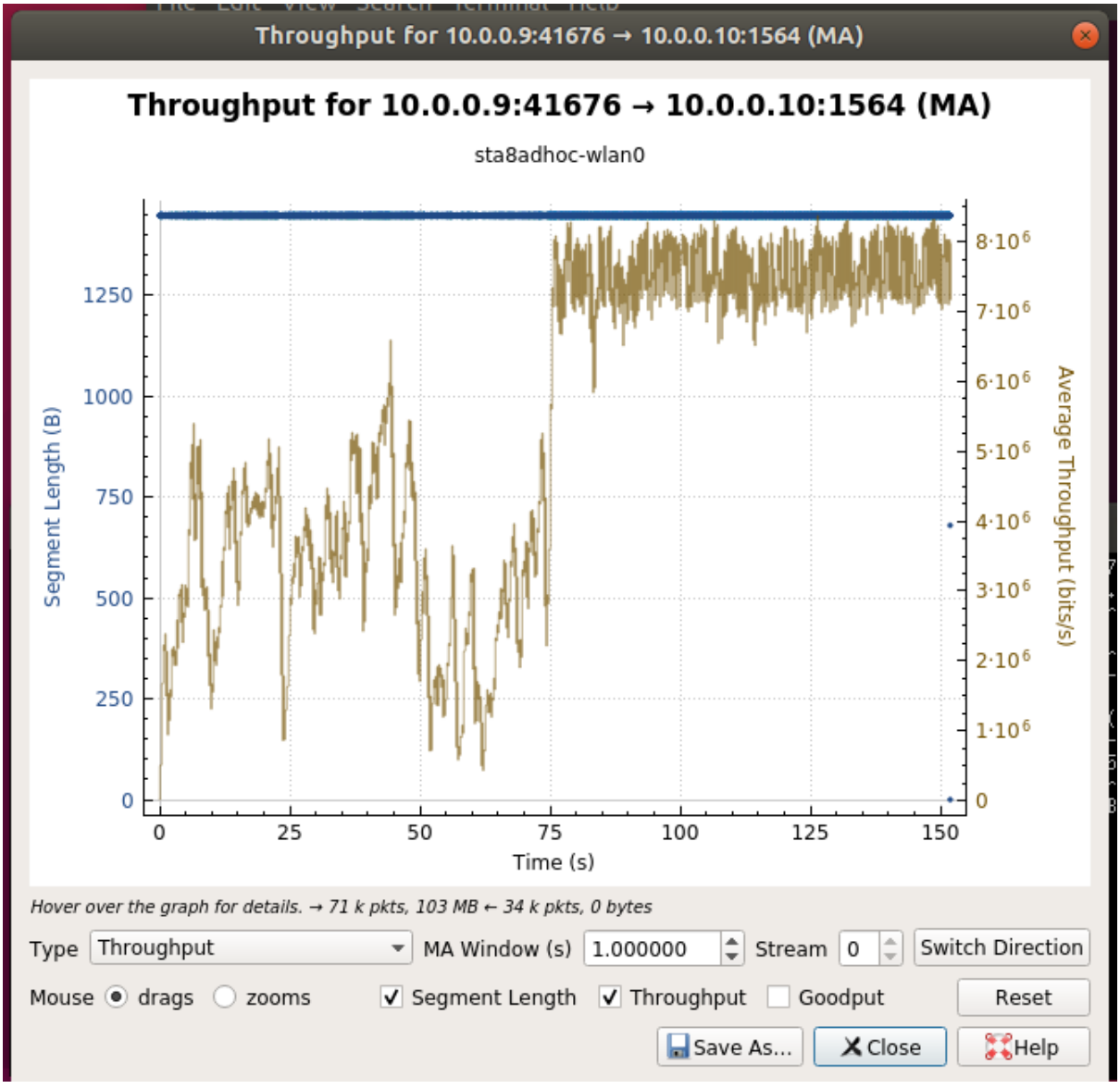
- Below is the screenshot of Average Throughput graph of transfer data between server – sta6adhoc and client sta5adhoc on port 1163:



In the above TCP transfer, 100MB of data is transferred in 134.0 seconds.

So, average throughput of that transfer is: **6.26** Mbits/sec.

- Below is the screenshot of Average Throughput graph of transfer data betweenserver – sta9adhoc and client sta8adhoc on port 1564:



In the above TCP transfer, 100MB of data is transferred in 157.3 seconds.

So, average throughput of that transfer is: **5.33**Mbits/sec.

The snapshots that have been attached, give an example of the system information that is required in the given adhoc stations.

- It is necessary to have speed, MAC address, IPv6 address, and username position for the given management.

- The calculated packet loss is found to be null.

iv. Analysis and discussion

- One can make sure that the level of the packet reaches the high target networking system as well as the ability of system management properly.

- The components used for this plan are Mininet, Adhoc networking, and WIFI networking (Castanheira, 2018)

- Two adhoc protocols define olsr, babel, and networking systems.

- Regular wireless networks cannot is inferior to that wireless adhoc networks.

c. Task3

i. Methodology

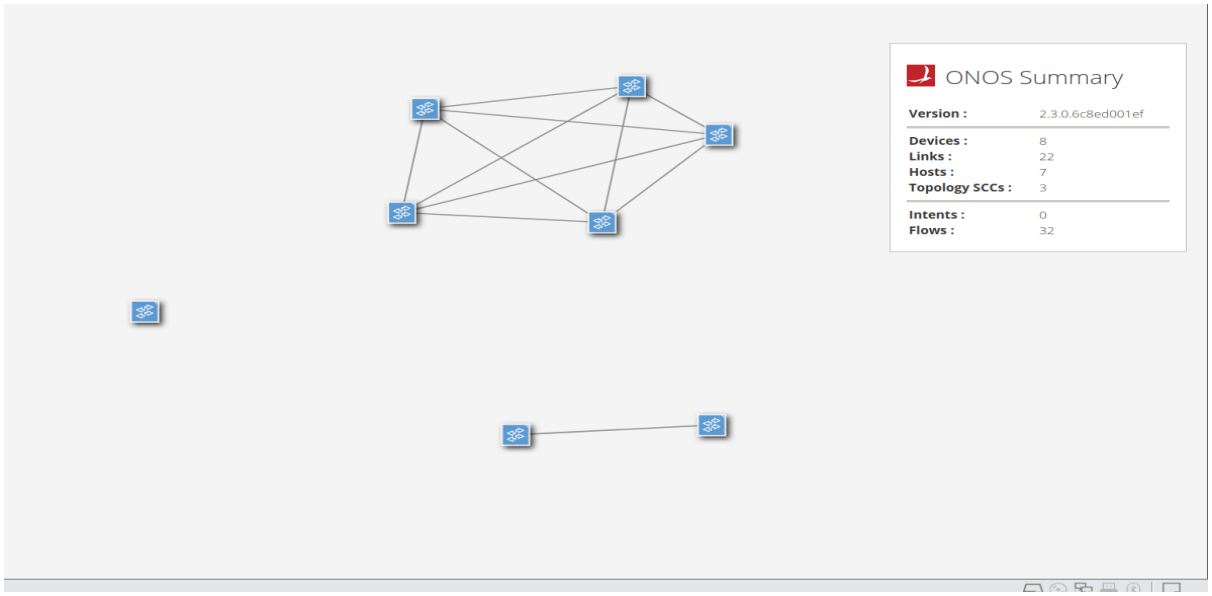
The system based on the university of Hertfordshire python scripture is used to emulate the environment. In between the server/UDP and host a UDP flow occurs which can be summed up to a total of 480. This should be used properly in the system management method so that the configuration command is dealt with with ease. In task 3, the ONOS controller can be applied (Bannour *et al.,* 2018). This methodology can control all the major operations for this task. According to the plan, two of the buildings and the server room should be connected where a separate VLAN and switches present in each building are examined thoroughly.

ii. Experiments

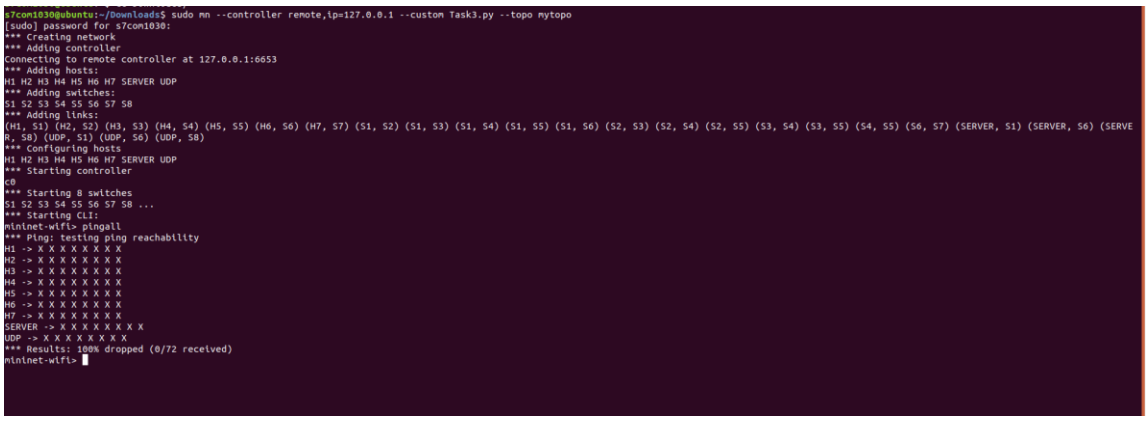
Objective of the situation can be maintained by the three major experiments such as focusing on the TCP communication to maintain coordination between the clients and server. Secondly, application of IPERF tool to ensure data transfer and traffic capture between clients and servers. Thirdly, application of Wireshark tools to ensure capture between the TCP traffic. There is the requirement of multicasting video system requirement to have design and management of operations (Arbettu et al. 2016). VLAN Functionality must be maintained to have access on each of channel. RDP connection and Wireshark packet capture must be the central focus of each of the operations.

iii. Evidence and Screenshots

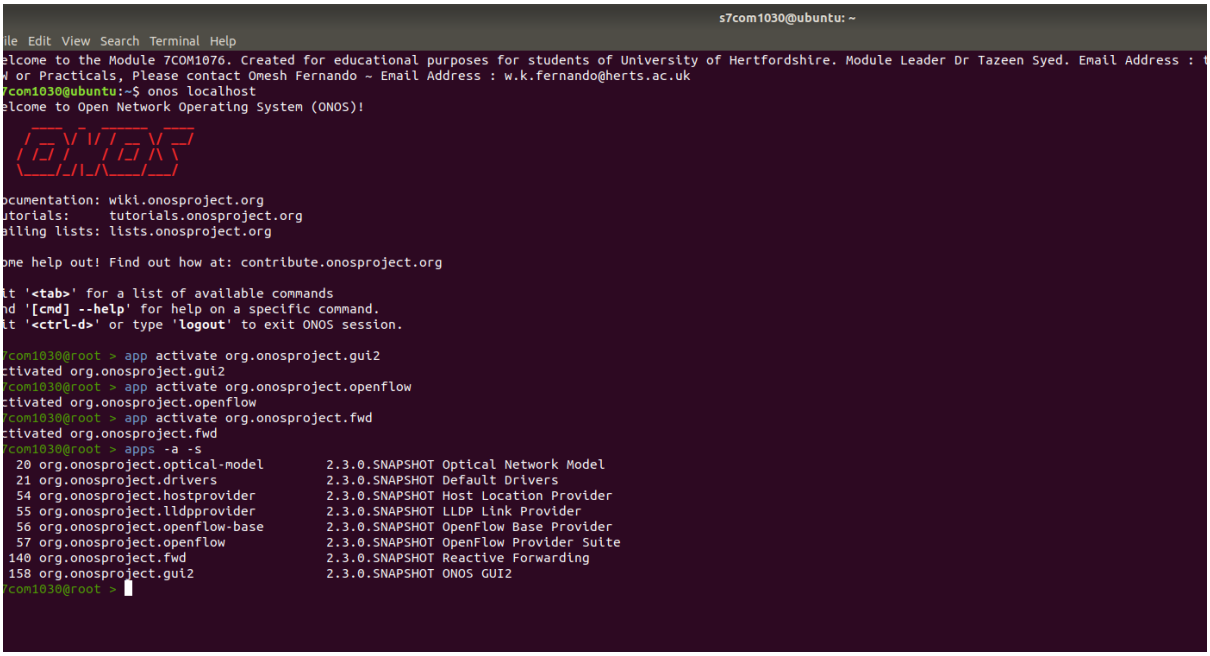
- The below mentioned screen shot provide clear indication of TCP communication to maintain server control.



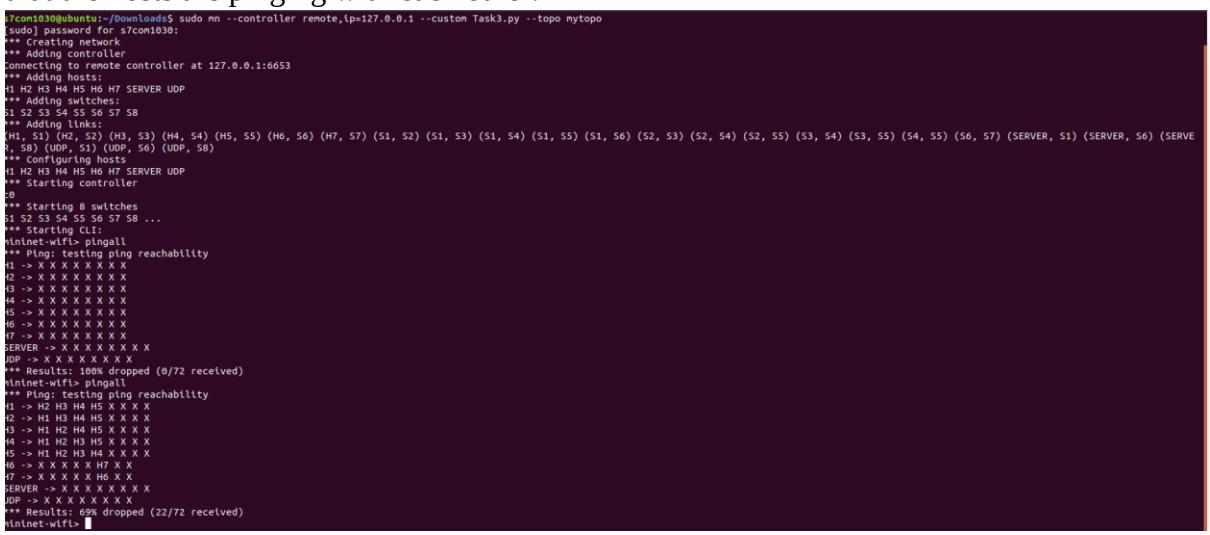
- Following are the screenshots for VLAN Functionality:
  - As we can see in the below screenshot that the hosts are not pinging with each other.



  - To overcome this, open the new terminal and write the command onos localhost which will open onos cli.
    Now in onos cli write the following commands:



- After running all the commands now again check the pinging of the hosts and we can see that the hosts are pinging with each other.



Based on the scenario lecturer room of each of the building must be separated and connection is done through the ring or grid topology (Arbettu et al. 2016). Therefore, before implementing and networking structure, it is required to have the Ping Command to test the networking system and control.

Task: 4

1.

Name	IP Address	Room	MAC	VLAN
H1	192.168.111.17/24	Lecture Room A	00:00:10:20:18:10	400
H2	192.168.111.18/24	Lecture Room B	00:00:10:20:18:11	400
H3	192.168.111.19/24	Lecture Room C	00:00:10:20:18:12	400
H4	192.168.111.20/24	Lecture Room D	00:00:10:20:18:13	400
H5	192.168.111.21/24	Lecture Room E	00:00:10:20:18:14	400
H6	192.168.111.22/24	Lecture Room F	00:00:10:20:18:15	500
H7	192.168.111.23/24	Lecture Room G	00:00:10:20:18:16	500
Server	20.0.0.1/8	Server Room	00:00:10:20:18:17	N/A
UDP	40.0.0.1/8	Server Room	00:00:10:20:18:18	N/A

2. Below are the commands which are used to configure the route between the server – SERVER and the client – H1 :

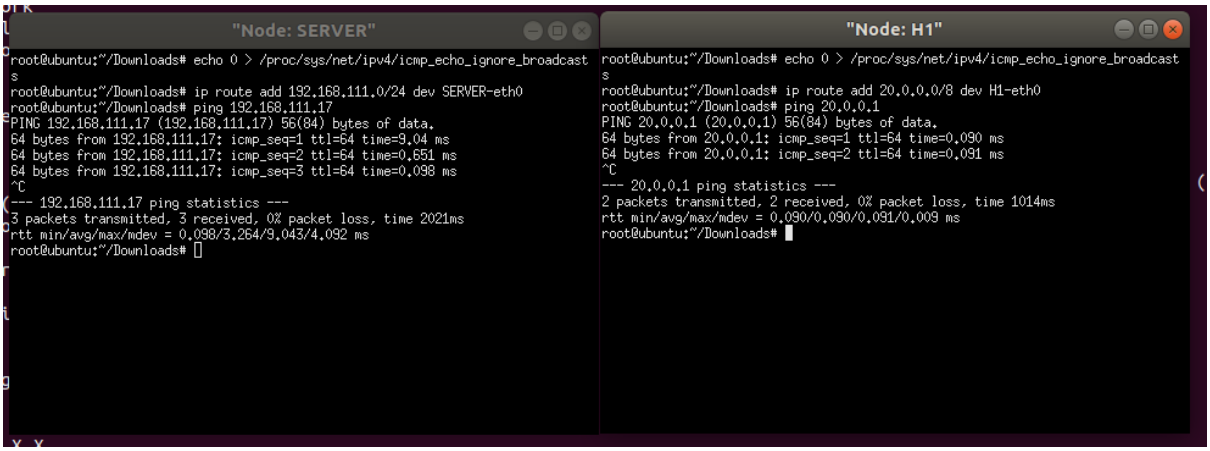
On the server side write the following commands:

- echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
- ip route add 192.168.111.0/24 dev SERVER-eth0
- ping 192.168.111.17

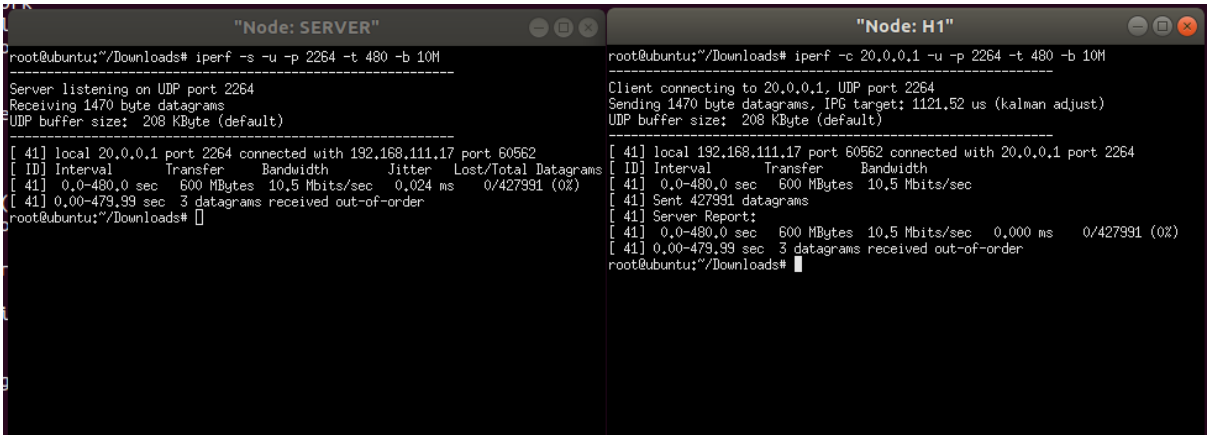
On the client side write the following commands:

- echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
- ip route add 20.0.0.0/8 dev H1-eth0
- ping 20.0.0.1.

Below screenshot shows that the route between the server and the client configured successfully:

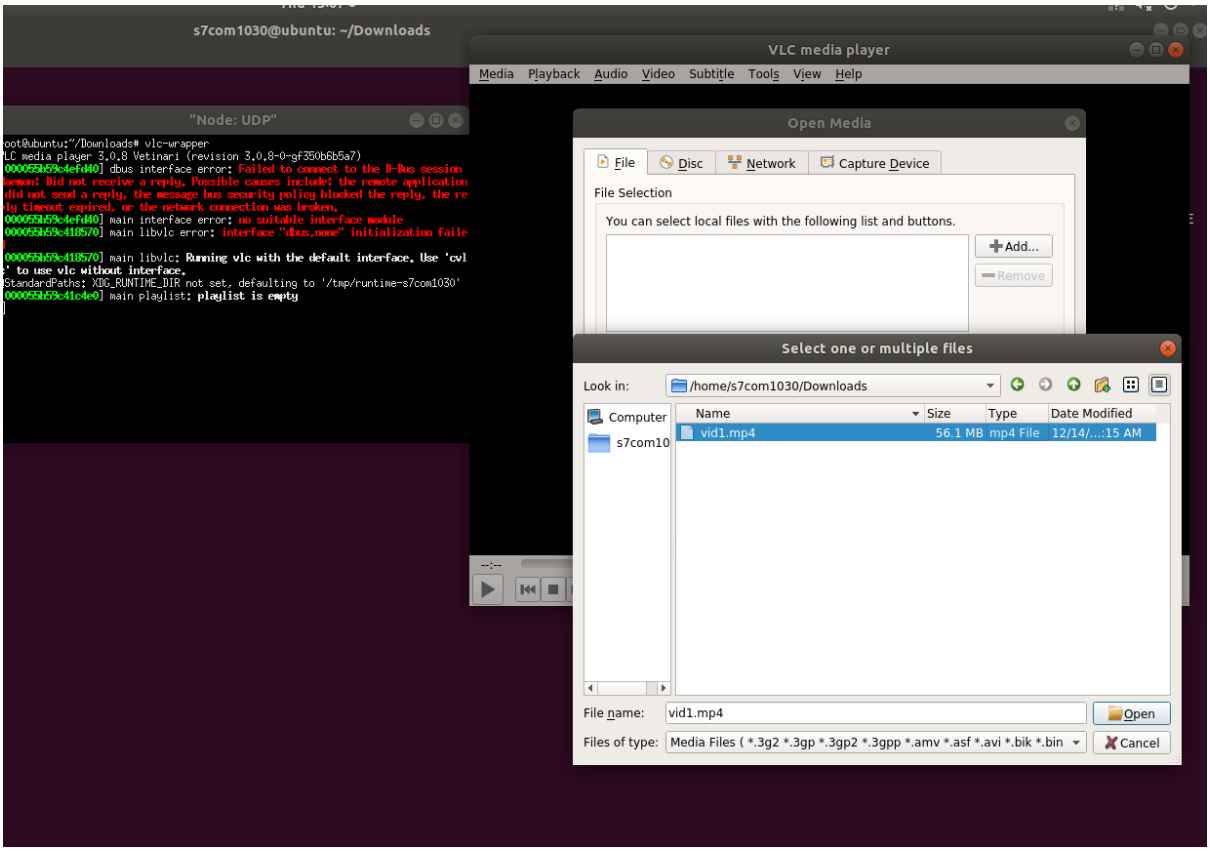


3. Below is the screenshot of confirming the route between SERVER and client-H1 on port 2264:

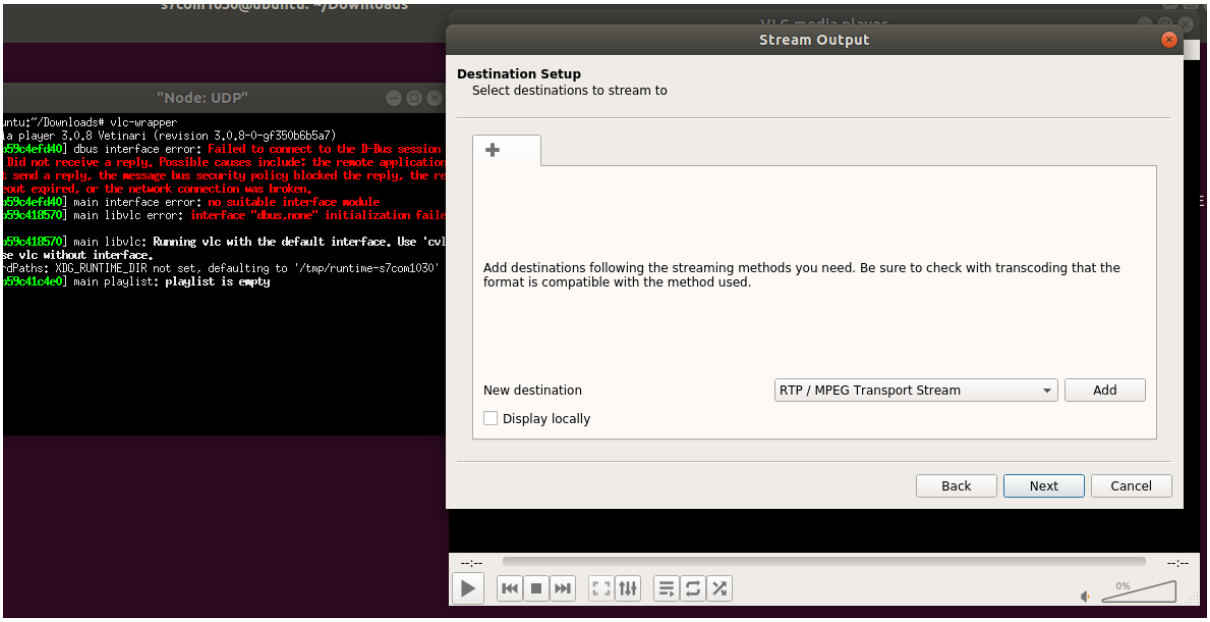


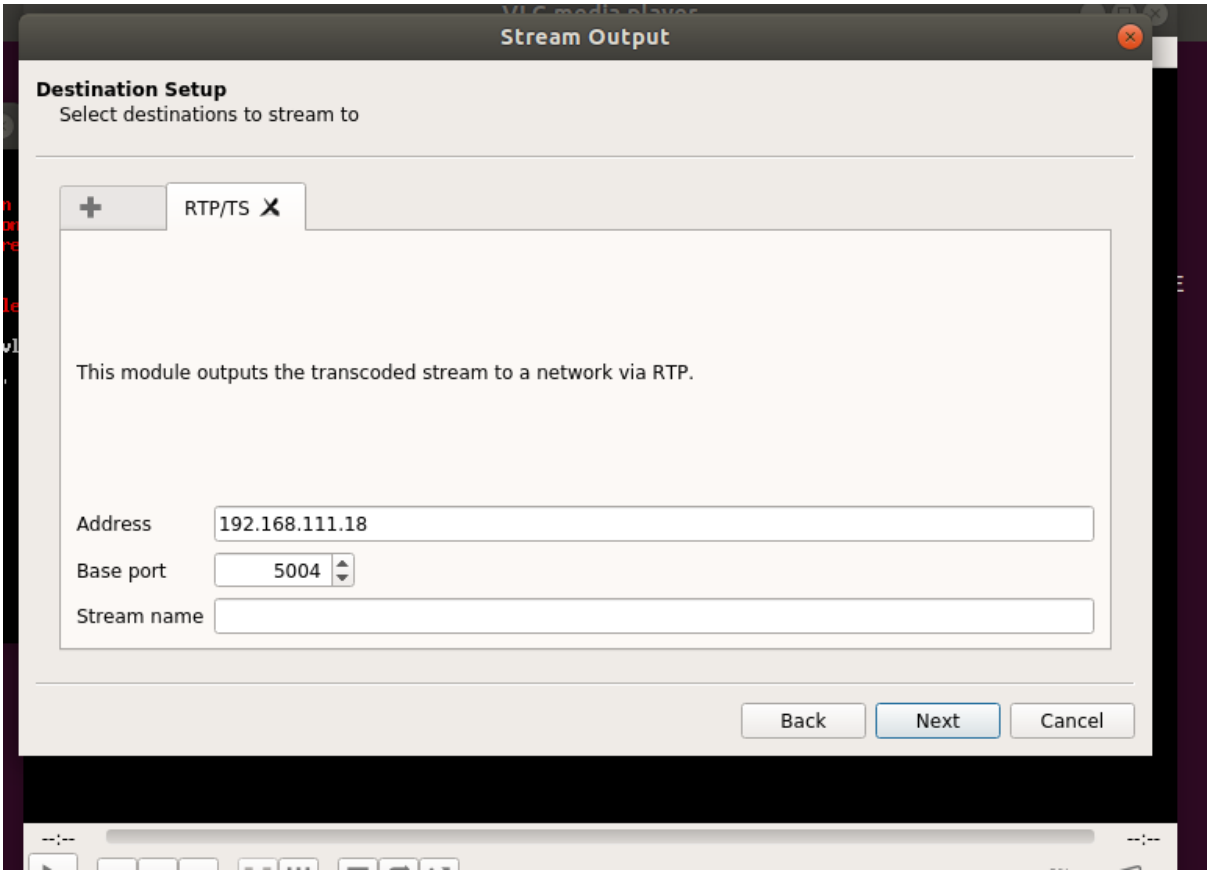
4. Below is the screenshot of video streaming of the video on both Host and Servers VLC players:

- First of all, on the server cli write the command **vlc-wrapper** to open the VLC media player and then in VLC media play go to Media and then select Stream option and in the Stream, option click on Add and then add the video file and then click on Stream.

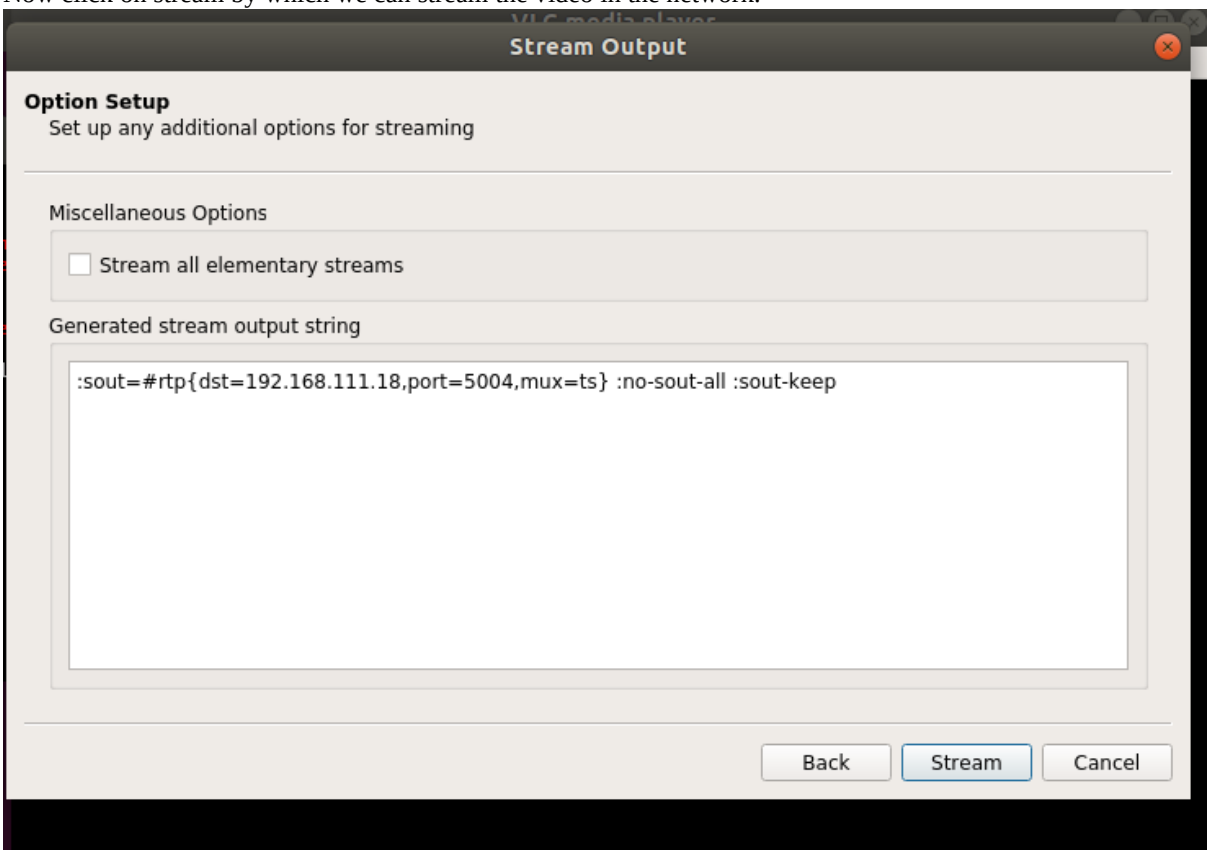


- Now in the Destination Setup, select the new destination as **RTP/MPEG Transport Stream** and then click on Add button.
  In the next window, write the IP address of the client on which we have to run the video and then click on Next.

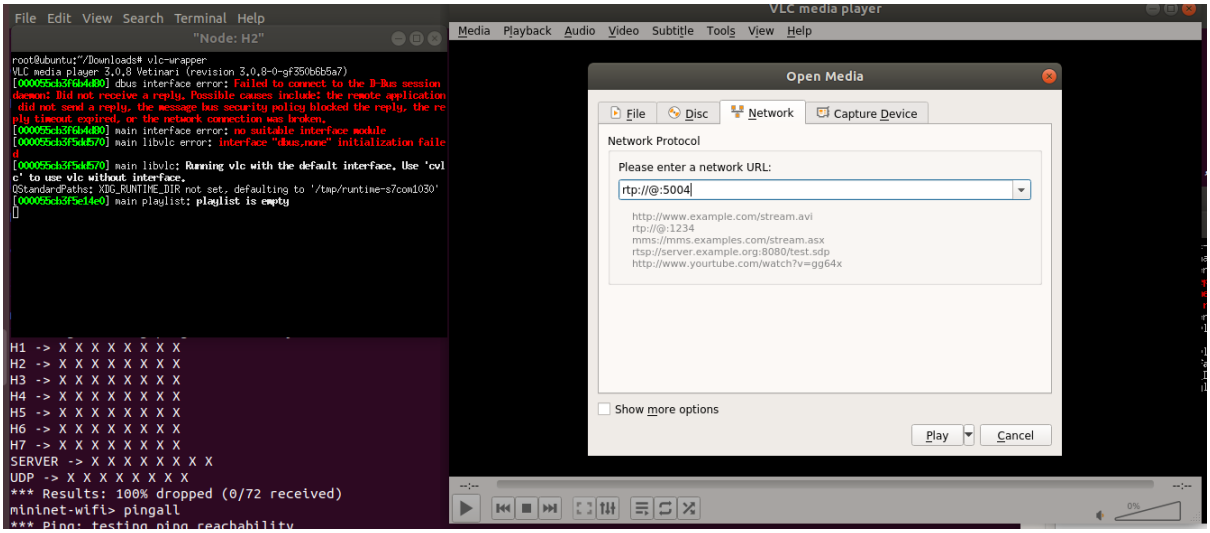

"Node: UDP"
Stream Output
Destination Setup
Select destinations to stream to
Add destinations following the streaming methods you need. Be sure to check with transcoding that the format is compatible with the method used.
New destination
RTP / MPEG Transport Stream
Add
Display locally
Back
Next
Cancel


Stream Output
Destination Setup
Select destinations to stream to
RTP/TS
This module outputs the transcoded stream to a network via RTP.
Address
192.168.111.18
Base port
5004
Stream name
Back
Next
Cancel

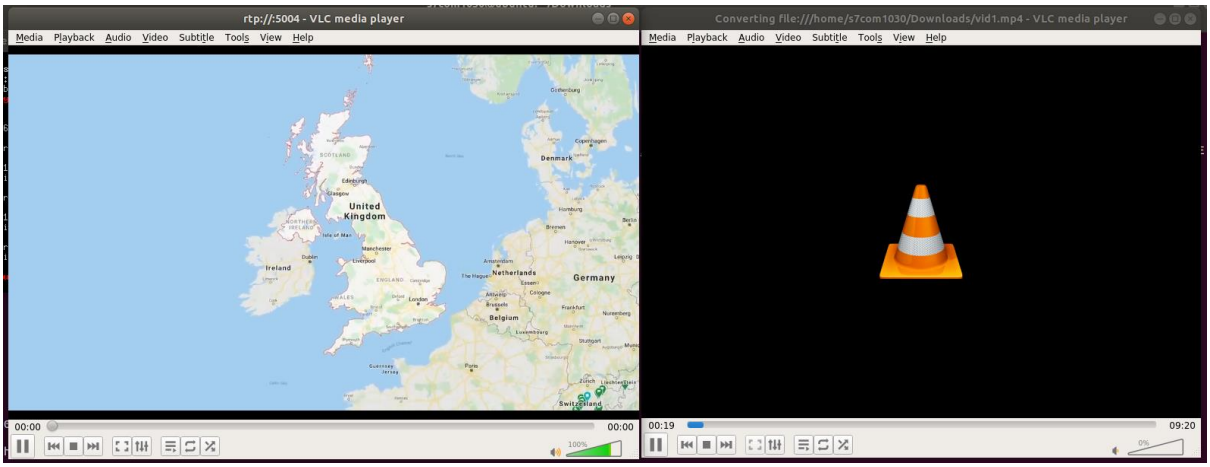
- Now click on stream by which we can stream the video in the network.



- Now write the command **vlc-wrapper** on the client cli and then in the VLC media player open the network stream from the media option and then write the network URL and then click on Play.



- As you can see our video file is running on the client vlc media player.



The ping command is the simplest and most common command for completing basic connectivity for testing. This process helps the moderator to display all the connectivity between all the nearest hosts to tell connects between each of the system.

iv. Analysis and discussion

Based on the above mentioned finding, it has been identified that significant delay in the trafficking can cause potential issue for networking management. The networking system must be well designed by the application of ping command to host communicates in realistic networking environment. The ping command work in the mininet by finding out the hosts. MAC address of the second can be controlled very significantly through the evaluation of the first host and pushes down a flow entry (Secci *et al.,* 2019). Creation of simple network of two hosts and one switch is the main focus of the system.

3. Conclusion:

Thus, it can conclude that main focus of the wireless network is to supporting several servers for video streaming and other distributed applications. UDP connection network is the simplest way to have performance on the operation. Application of multicast streaming & SDN network is useful to establish communication. First step involve application of mininet. Secondly, setup of ad hoc stations is very effective for operation. Finally, utilizing the ONOS tool is the network trafficking in the server and client is very important.

Reference list:

Arbettu, R. K., Khondoker, R., Bayarou, K., & Weber, F. (2016, September). Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers. In *2016 17th International telecommunications network strategy and planning symposium (Networks)* (pp. 37-44). IEEE. https://publica.fraunhofer.de/bitstreams/e7e33b28-6589-41e7-b81e-a76b9c87bf67/download

DeCusatis, C., Carranza, A., & Delgado-Caceres, J. (2016, May). Modeling Software Defined Networks using Mininet. In *Proc. 2nd Int. Conf. Comput. Inf. Sci. Technol. Ottawa, Canada* (No. 133, pp. 1-6). http://avestia.com/CIST2016_Proceedings/files/paper/133.pdf

Gupta, N., Maashi, M. S., Tanwar, S., Badotra, S., Aljebreen, M., & Bharany, S. (2022). A Comparative Study of Software Defined Networking Controllers Using Mininet. *Electronics*, *11*(17), 2715. https://www.mdpi.com/2079-9292/11/17/2715/pdf

Muqaddas, A. S., Bianco, A., Giaccone, P., & Maier, G. (2016, May). Inter-controller traffic in ONOS clusters for SDN networks. In *2016 IEEE international conference on communications (ICC)* (pp. 1-6). IEEE. https://core.ac.uk/download/pdf/234909201.pdf

Van Tu, N., Hyun, J., & Hong, J. W. K. (2017, September). Towards onos-based sdn monitoring using in-band network telemetry. In *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 76-81). IEEE. https://pdfs.semanticscholar.org/a50e/4d6c42da4a808610fcd9cb8b93cf73571bc1.pdf

Hong, F.T., Huang, X., Li, W.H. & Zheng, W.S., (2020, August). Mini-net: Multiple instance ranking network for video highlight detection. In *European Conference on Computer*

Vision (pp. 345-360). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-030-58601-0_21

Monika, P., Negara, R.M. & Sanjoyo, D.D., (2020). Performance analysis of software defined network using intent monitor and reroute method on ONOS controller. *Bulletin of Electrical Engineering and Informatics*, *9*(5), pp.2065-2073. https://www.beei.org/index.php/EEI/article/view/2413

Gupta, N., Maashi, M.S., Tanwar, S., Badotra, S., Aljebreen, M. & Bharany, S., (2022). A Comparative Study of Software Defined Networking Controllers Using Mininet. *Electronics*, *11*(17), p.2715. https://www.mdpi.com/2079-9292/11/17/2715

Pang, L., Yang, C., Chen, D., Song, Y. & Guizani, M., (2020). A survey on intent-driven networks. *IEEE Access*, *8*, pp.22862-22873. https://ieeexplore.ieee.org/abstract/document/8968429

Hedges, L., Mey, A., Laughton, C., Gervasio, F., Mulholland, A., Woods, C. & Michel, J., (2019). BioSimSpace: An interoperable Python framework for biomolecular simulation. *Journal of Open Source Software*, *4*(43). https://nottingham-repository.worktribe.com/index.php/output/3611779/biosimspace-an-interoperable-python-framework-for-biomolecular-simulation

Castanheira, T.M.A., (2018). Simulation of mobile edge-cloud applications using Mininet-WiFi. https://repositorio-aberto.up.pt/bitstream/10216/118829/2/312178.pdf

Bannour, F., Souihi, S. & Mellouk, A., (2018, December). Adaptive state consistency for distributed onos controllers. In *2018 IEEE Global Communications Conference (Globecom)* (pp. 1-7). IEEE. https://ieeexplore.ieee.org/abstract/document/8647168

Secci, S., Diamanti, A., Vilchez, J.M.S., Bah, M.T., Vizzarreta, P., Machuca, C.M., Scott-Hayward, S. & Smith, D., (2019). *Security and Performance Comparison of ONOS and ODL controllers* (Doctoral dissertation, ONOS). https://hal.archives-ouvertes.fr/hal-03188550/document