

# Implementation of SDN in a Data Center Network

Gokul Anudeep Pokuri, Jayashree Naidu Kotte

Department of Electrical Engineering, San Jose State University, San Jose, California 95192

## Introduction

Implementation of SDN in a Data Center Network enables centralized network monitoring and higher routing efficiency by separating control and data planes thus improving the usage of bandwidth, reduction of congestion and less packet drop rate.

### Software Defined Networking (SDN);

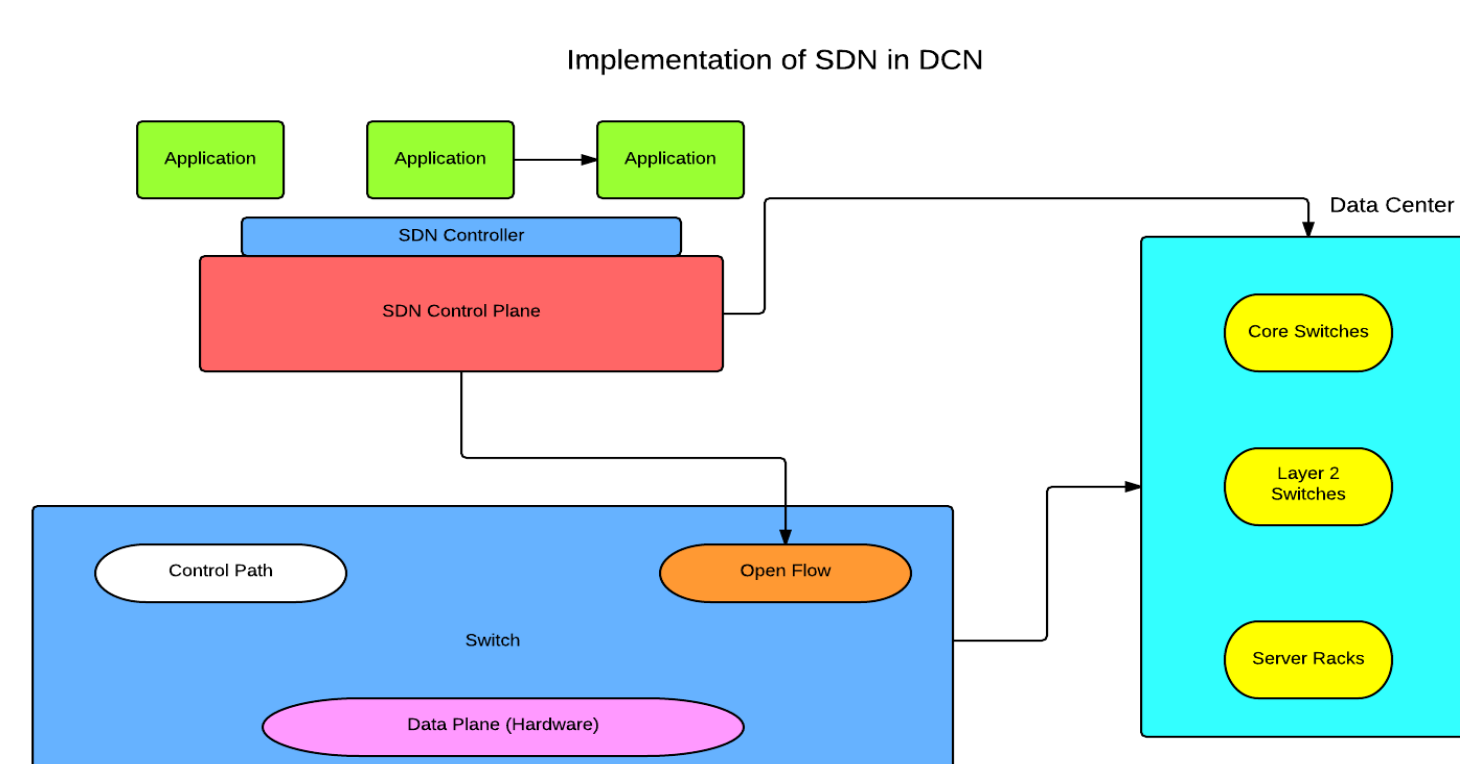
- Separates the control and data planes,
- Controls the entire network infrastructure using SDN controller.
- OpenFlow protocol is used by the SDN controller to communicate with vSwitches to forward packets in the network utilizing the entire bandwidth available.

### Data Center Network (DCN):

- Is a physical or virtual warehouse of data
- DCN is the heart of every industry and has a three-tier architecture.
- Three-tier architecture comprises of core L3 switches, aggregate L2/L3 switches and server racks
- DCN comes in broadly two architectures,
  1. Server-routed network design
  2. Switch-routed network design

The objective of this work is to draw inferences of routing speed, latency, scaling and bandwidth utilization by implementing SDN on a complex DCN design.

## SDN in a Data Center Network



The integration of SDN on a data center network is depicted in the above figure.

SDN abstracts the network resources from the physical infrastructure and divides the infrastructure into two separate entities. These entities are the control and data planes. Data plane acts merely as a packet forwarding backplane, whereas the control plane is responsible for all the decisions made to formulate paths and routing tables for forwarding packets. The entire process is accomplished using a communication medium called OpenFlow protocol.

## Key References

- [1] "Computer and Communication Networks" by Nader F. Mir.
- [2] White paper on "SDN meets the real world implementation benefits and challenges" by Gigaom. <http://www.nuagenetworks.net/wp-content/uploads/2014/11/Gigaom-Research-SDN-Meets-the-Real-World-Final.pdf>
- [3] White paper on "An Introduction to Software Defined Networking" by Citrix. [https://www.citrix.com/content/dam/citrix/en\\_us/documents/oth/sdn-101-an-introduction-to-software-defined-networking.pdf](https://www.citrix.com/content/dam/citrix/en_us/documents/oth/sdn-101-an-introduction-to-software-defined-networking.pdf)
- [4] "SDN: Software Defined Networks" by Thomas D. Nadeau & Ken Gray.

## Project Design Implementation

### Key Tools

- VMware Workstation: Hypervisor
- Mininet: Network emulator
- POX and RYU controllers: OpenFlow controllers
- Python scripting language

Virtualization play a key role in this project. SDN enables application of virtualization principles on network infrastructure by abstracting network resources, pooling and automating them to outshine the limitations of DCN architectures.

The procedure of virtualization aided by SDN is achieved by installing a VMware workstation such as VMware fusion, VM virtualbox etc., and installing Mininet on the VM.

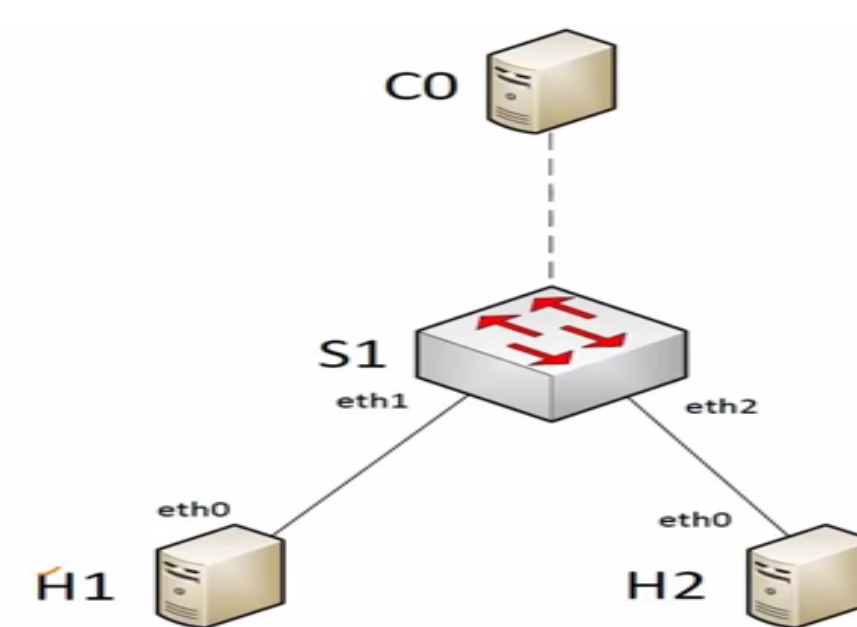
Mininet is a network emulator that has inbuilt virtual switches, routers and openflow controller in it. Mininet also allows creation of custom topologies using python scripts. In mininet CLI we develop a custom tree topology using python scripts for DCN and control the decision making by SDN controller. We can also make use of Mininet commands to create custom tree topology of our choice.

Below is the command that generates a network with 2 hosts and one switch.

Command: `$sudo mn -topo=single,2`

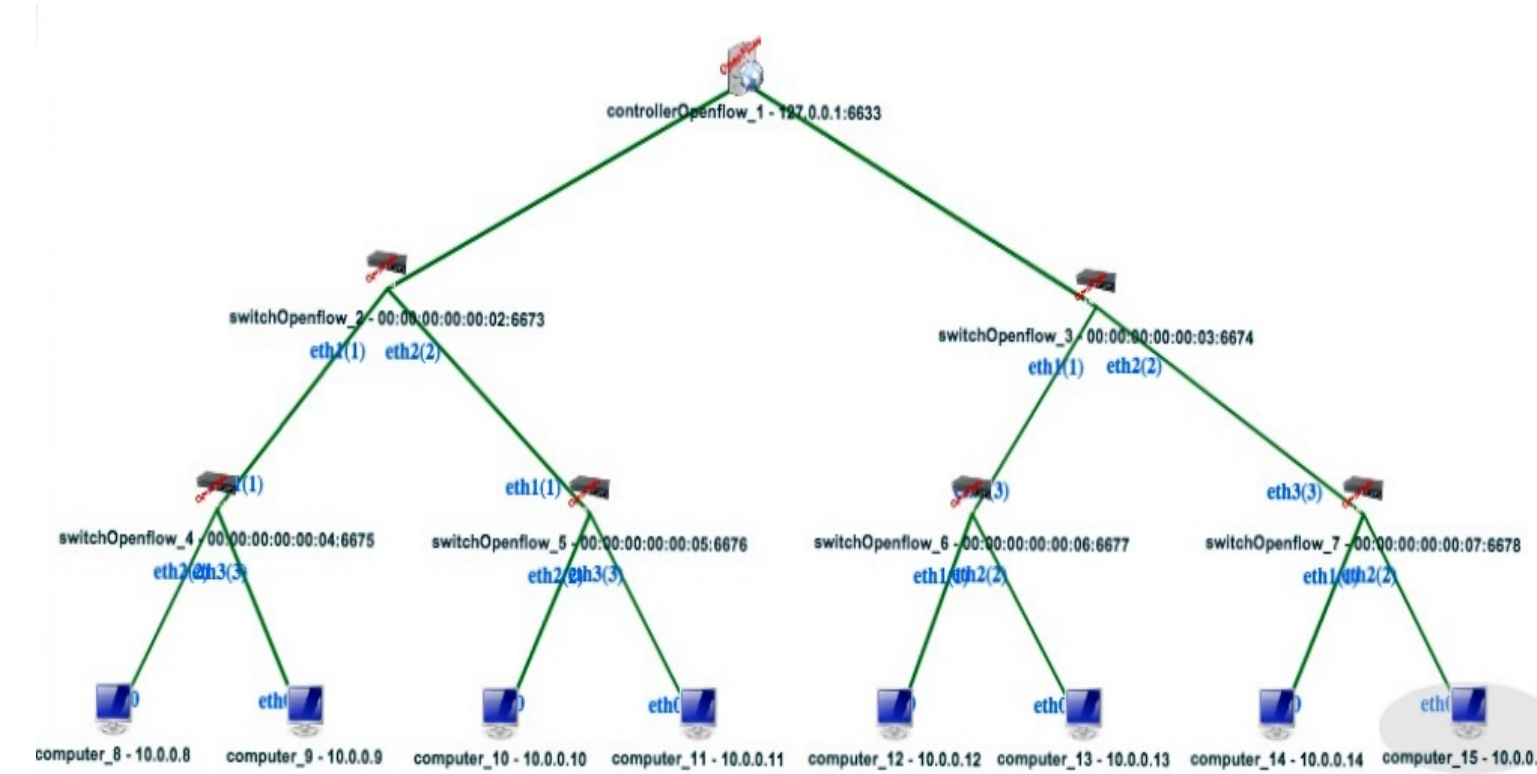
The below figure is a result of the command used in mininet.

### Basic tree topology with 2 hosts



The same topology can be extended to create a tree topology with 8 hosts using the following command, the figure shows the 8 host tree topology. Command: `$sudo mn -topo=tree,3,2`

### Tree topology of a DCN with 8 hosts

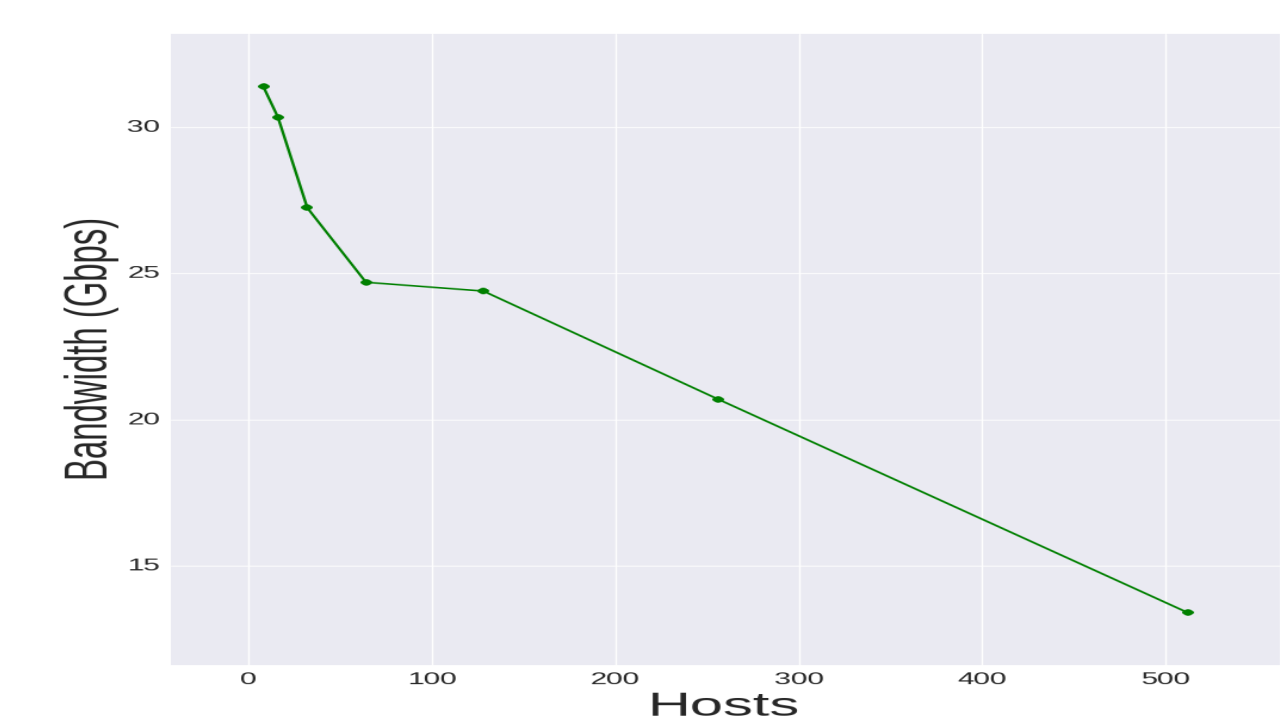


Likewise we scaled the network to 512 hosts.

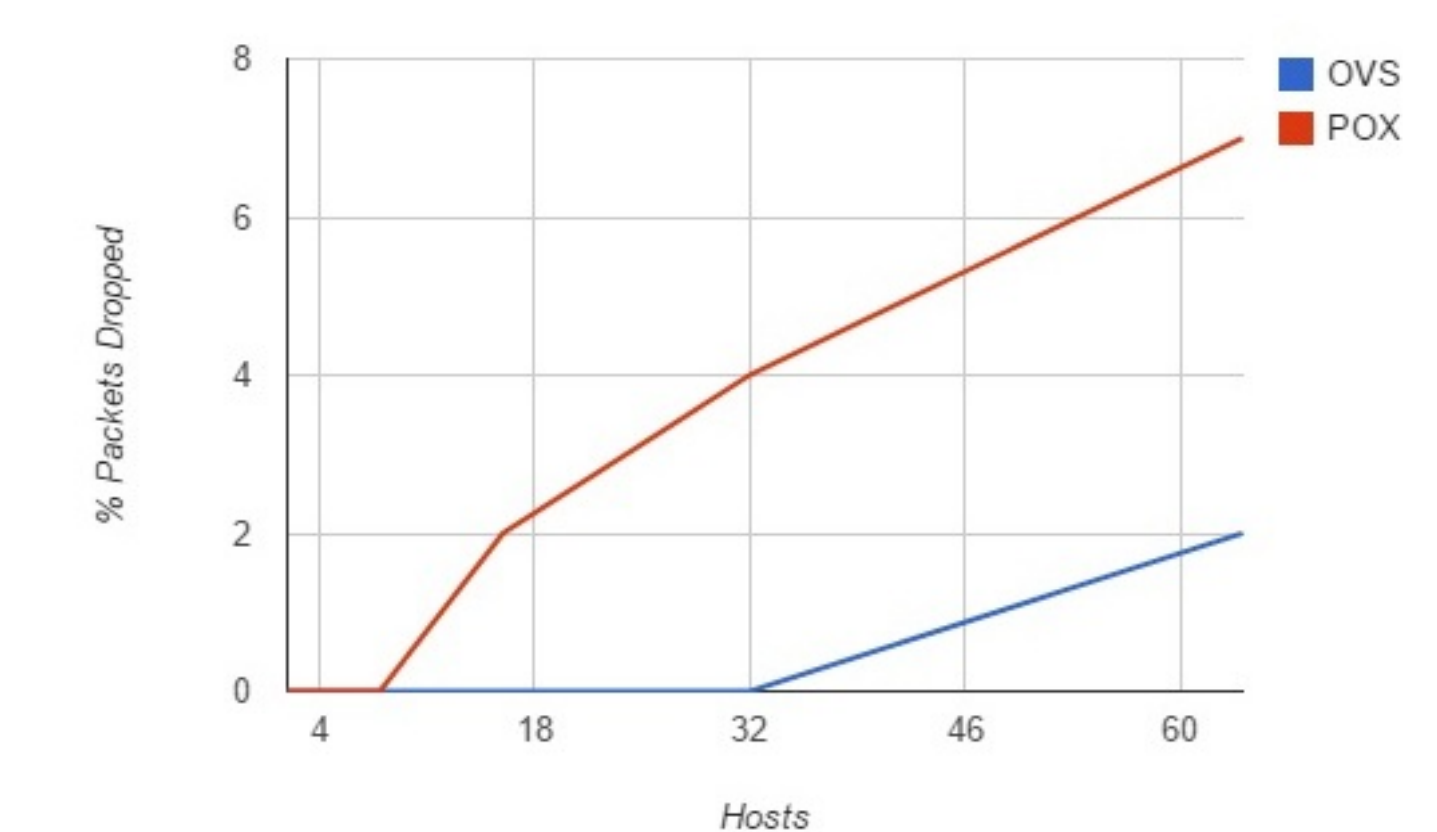
## Results

Performance metrics such as bandwidth, packet drop rate and latency have been compared and inferences were drawn. The below graphs gives us an estimation of the performance of SDN.

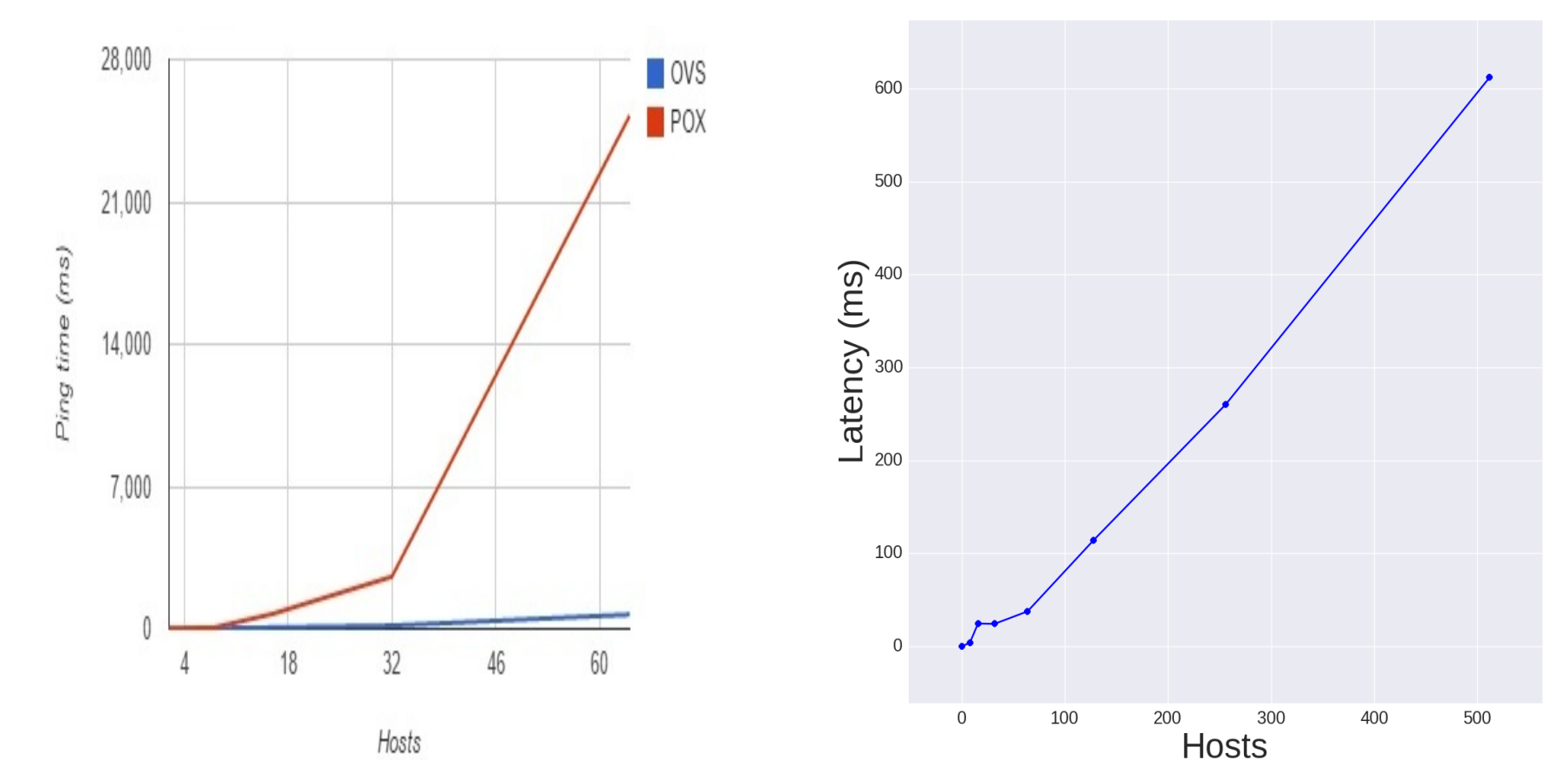
### Bandwidth vs. number of hosts



### Packet drop rate vs. number of hosts



### Latency (Ping time) vs. number of hosts



These graphs provide insight into the efficiency of SDN.

## Conclusions

We have implemented SDN on a data center network making use of virtualization principles. We have looked at how the network scales from 8 hosts to 512 hosts in terms of latency, packet drop rate and bandwidth. From the results it can be safely concluded that as the number of hosts increase the bandwidth available decreases, latency between the host and packet drop rate increases accordingly. These results are in accordance with expected results at the beginning of the project and show the efficiency of SDN over traditional network infrastructure.

## Acknowledgments

We thank Nader F. Mir for providing invaluable expertise and unwavering support which aided us in the completion of this project.

## For further information

Please contact [jayashree.kotte@gmail.com](mailto:jayashree.kotte@gmail.com). Python code, simulation files and topology diagrams are available upon request.