**Software-Defined Networking Assignment 1 2019-20:**

**Design and create a simple Mininet network with a single controller and a number of switches and hosts.**

## Description

Write a Mininet script in Python to create a network of 8 switches and 12 hosts, and a remote controller, according to the following topology diagram.

**S1**

**S3**

**S4**

**S5**

**H1**

**H2**

**H3**

**H4**

**H5**

**H6**

**S2**

**S6**

**S7**

**S8**

**H7**

**H8**

**H9**

**H10**

**H11**

**H12**

The switches should be Open vSwitch instances, and the remote controller will be ONOS. Make sure your script names the switches and hosts as they are named in the topology diagram. After the network is created, the user should be presented with the Mininet CLI.

Do a sample run of your script. In the Mininet CLI, test that all hosts can ping each other successfully – take a screenshot of the command you used to start the script, the resulting output of Mininet, the pingall command and the output of that. Take a screenshot of the ONOS GUI after it has detected the switches and hosts.

You must submit a .zip file (named <student-number>-<firstname.surname>-assign1.zip) include only the following files:

* sdntopo1.py – a commented Python script to create the Mininet topology. Include a comment in the file to assert that you are the author of the script.
* screenshot1.jpg – a screenshot of ONOS showing the topology (including switches with DPIDs and hosts with IP addresses).
* screenshot2.jpg – a screenshot showing how you started Mininet with your script, the output as Mininet starts up, the pingall command, and successful output from pingall showing that all hosts are reachable.

## Other Information

* Submission is through Canvas (<http://cit.instructure.com>) only.
* Assignment value: 15%
* Submission date: 8 Nov 2019

## Marking

* Correctly following submission instructions - 7% (5 marks for correct naming of files, 3 marks for correct formats – py, jpg & zip)
* Scripting style – 18%
  + Authorship assertion
  + Useful comments
  + Consistent indentation, and use of white space for readability
  + No unnecessary code
  + Use of loops
  + Use of constants or parameters instead of hard-coded values in functions
* Correctly building topology (switches and hosts) – 20%
  + No. of switches & hosts correct
  + Names of switches, hosts and controller correct
  + Links created between switches, and between switches and hosts
* Correctly identifying remote controller – 10%
* Correct Mininet command – 5%
* Output showing successful Mininet startup – 5%
* Mininet CLI – 5%
* Pingall successful – 10%
* Topology visible at controller (switches and hosts) – 20%
  + All switches and hosts present
  + All links shown
  + DPIDs shown
  + Host IP addresses shown

[Changes for next time:

* …

]