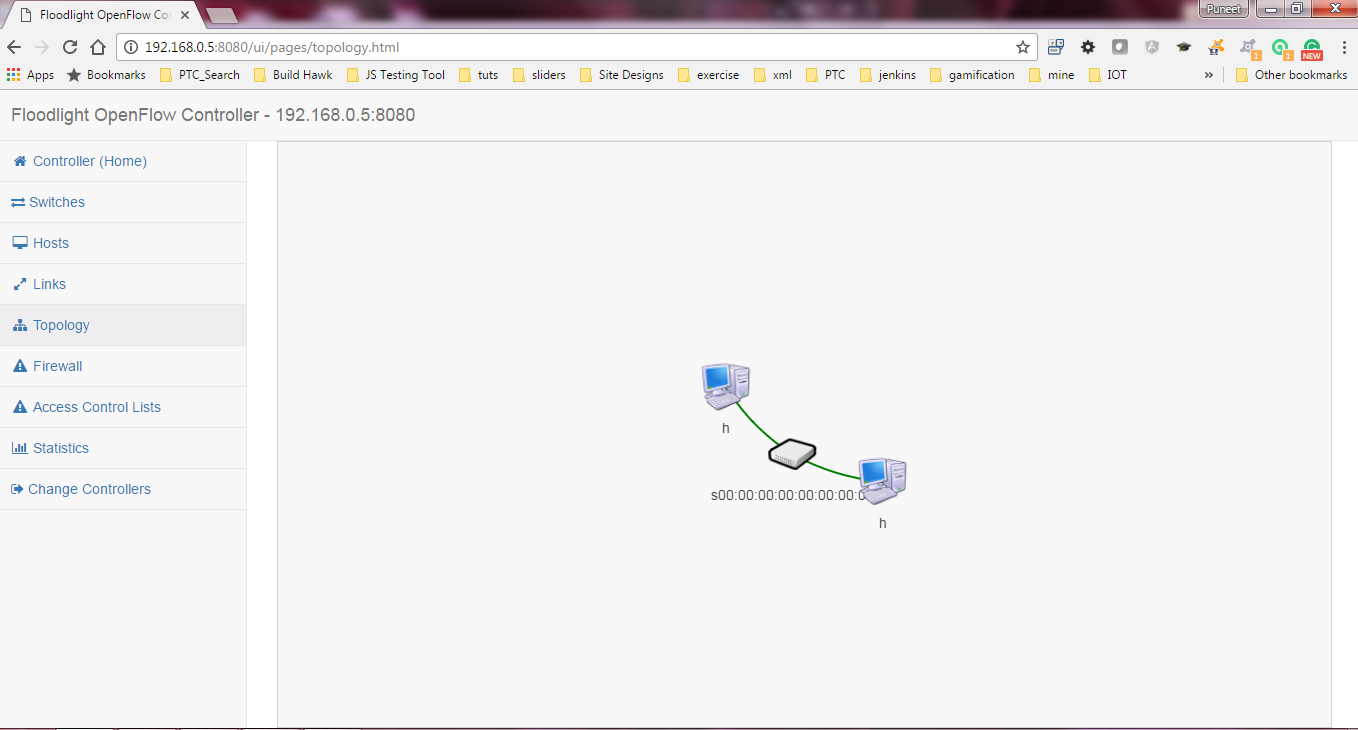
**Implementation and Testing**

To test our so far work we have created a custom topology using Mininet Virtual Network. The topology is very basic and consists of two hosts and one switch. Please refer the screenshot below:



We plan to create various different types of topologies further in our project. Since Mininet supports Python, we have written our custom topology in python. Following is the python script we have used to create above custom topology:

from mininet.node import CPULimitedHost

from mininet.topo import Topo

from mininet.net import Mininet

from mininet.log import setLogLevel, info

from mininet.node import RemoteController

from mininet.cli import CLI

from mininet.link import TCLink

class GenericTree(Topo):

"""Simple topology example."""

def build( self, depth=1, fanout=2 ):

# Numbering: h1..N, s1..M

self.hostNum = 1

self.switchNum = 1

def build( self, depth=1, fanout=2 ):

# Numbering: h1..N, s1..M

self.hostNum = 1

self.switchNum = 1

# Build topology

self.addTree(depth, fanout)

def addTree( self, depth, fanout ):

"""Add a subtree starting with node n.

returns: last node added"""

isSwitch = depth > 0

if isSwitch:

node = self.addSwitch( 's%s' % self.switchNum )

self.switchNum += 1

for \_ in range( fanout ):

child = self.addTree( depth - 1, fanout )

self.addLink( node, child )

else:

node = self.addHost( 'h%s' % self.hostNum )

self.hostNum += 1

return node

def run():

c = RemoteController('c')

# Change the args of GenericTree() to your desired values. You could even get them from command line.

net = Mininet(topo=GenericTree(depth=2, fanout=3), host=CPULimitedHost, controller=None)

net.addController(c)

net.start()

# installStaticFlows( net )

CLI(net)

net.stop()

# if the script is run directly (sudo custom/optical.py):

if \_\_name\_\_ == '\_\_main\_\_':

setLogLevel('info')

run()