7. mobile stations move within the coverage are

import sys

from mininet.log import setLogLevel, info

from mn\_wifi.cli import CLI

from mn\_wifi.net import Mininet\_wifi

def topology(args):

"Create a network."

net = Mininet\_wifi()

info("\*\*\* Creating nodes\n")

h1 = net.addHost('h1', mac='00:00:00:00:00:01', ip='10.0.0.1/8')

sta1 = net.addStation('sta1', mac='00:00:00:00:00:02', ip='10.0.0.2/8')

sta2 = net.addStation('sta2', mac='00:00:00:00:00:03', ip='10.0.0.3/8')

ap1 = net.addAccessPoint('ap1', ssid='new-ssid', mode='g', channel='1',

position='45,40,0')

c1 = net.addController('c1')

info("\*\*\* Configuring propagation model\n")

net.setPropagationModel(model="logDistance", exp=4.5)

info("\*\*\* Configuring nodes\n")

net.configureNodes()

info("\*\*\* Associating and Creating links\n")

net.addLink(ap1, h1)

if '-p' not in args:

net.plotGraph(max\_x=200, max\_y=200)

if '-c' in args:

sta1.coord = ['40.0,30.0,0.0', '31.0,10.0,0.0', '31.0,30.0,0.0']

sta2.coord = ['40.0,40.0,0.0', '55.0,31.0,0.0', '55.0,81.0,0.0']

net.startMobility(time=0, mob\_rep=1, reverse=False)

p1, p2, p3, p4 = {}, {}, {}, {}

if '-c' not in args:

p1 = {'position': '40.0,30.0,0.0'}

p2 = {'position': '40.0,40.0,0.0'}

p3 = {'position': '31.0,10.0,0.0'}

p4 = {'position': '55.0,31.0,0.0'}

net.mobility(sta1, 'start', time=1, \*\*p1)

net.mobility(sta2, 'start', time=2, \*\*p2)

net.mobility(sta1, 'stop', time=12, \*\*p3)

net.mobility(sta2, 'stop', time=22, \*\*p4)

net.stopMobility(time=23)

info("\*\*\* Starting network\n")

net.build()

c1.start()

ap1.start([c1])

info("\*\*\* Running CLI\n")

CLI(net)

info("\*\*\* Stopping network\n")

net.stop()

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

setLogLevel('info')

topology(sys.argv)

8.Performance Evaluation of a Web Server in a Wireless Network(with one host acting as a web server and multiple mobile clients accessing the server)

from mininet.log import setLogLevel, info

from mn\_wifi.cli import CLI

from mn\_wifi.net import Mininet\_wifi

from mininet.node import Controller

from mininet.link import TCLink

import subprocess

def start\_wireshark(interface):

command = ['wireshark','-i',interface]

subprocess.Popen(command)

def web\_server\_topology():

"Create a wireless network with a web server and multiple clients."

net = Mininet\_wifi(controller=Controller,link=TCLink)

info('\* Adding controller\n')

net.addController('c0')

# Add an access point (AP)

ap1 = net.addAccessPoint('ap1', ssid='wifi-net', mode='g', channel='1', position='50,50,0',ip='10.10.0.0', range=100)

# Add a web server host (H1)

h1 = net.addHost('h1', ip='10.0.0.1')

# Add mobile stations (STA) as clients

sta1 = net.addStation('sta1', position='10,50,0',ip='10.10.0.1')

sta2 = net.addStation('sta2', position='20,50,0',ip='10.10.0.2')

sta3 = net.addStation('sta3', position='30,50,0',ip='10.10.0.3')

net.setPropagationModel(model="logDistance", exp=6)

# Configure nodes

net.configureWifiNodes()

# Add links between AP and H1 (web server)

net.addLink(ap1, h1)

# Add links between AP and stations (clients)

net.addLink(sta1, ap1,bw=10)

net.addLink(sta2, ap1,bw=20)

net.addLink(sta3, ap1,bw=30)

net.plotGraph()

net.startMobility(time=0)

net.mobility(sta1, 'start', time=1, position='10,30,0')

net.mobility(sta2, 'start', time=2, position='10,40,0')

net.mobility(sta1, 'stop', time=10, position='40,40,20')

net.mobility(sta2, 'stop', time=10, position='25,40,0')

net.mobility(sta3, 'start', time=3,position ='78,50,0')

net.mobility(sta3, 'stop', time=10,position = '10,40,0')

net.stopMobility(time=11)

net.start()

interface = 'ap1-wlan1'

start\_wireshark(interface)

# Configure web server on H1

info("\* Configuring web server on h1 \*\n")

h1.cmd('echo "<html><body><h1>Hello, Mininet!</h1></body></html>" > /tmp/index.html')

h1.cmd('python3 -m http.server 80 --directory /tmp &')

net.pingAll()

# Start CLI for interactive testing

info("\n\*\* Running CLI \*\*\n")

CLI(net)

# Stop the network

net.stop()

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

setLogLevel('info')

web\_server\_topology()

5. Wireless Control Traffic in Mininet-WiFi(wireshark)

from mininet.log import setLogLevel, info

from mininet.node import OVSSwitch, Controller

from mininet.cli import CLI

from mn\_wifi.net import Mininet\_wifi

import subprocess

def start\_wireshark(interface):

# Start Wireshark to capture traffic on the specified interface

command = ['wireshark', '-i', interface]

subprocess.Popen(command)

def create():

net = Mininet\_wifi(switch=OVSSwitch, controller=Controller)

net.addController('c0')

# Access points

ap1 = net.addAccessPoint('ap1', mode='g', position='20,20,10', range='20')

ap2 = net.addAccessPoint('ap2', mode='g', position='40,20,20', range='20')

# Stations

sta1 = net.addStation('sta1', position='10,10,10', range='5')

sta2 = net.addStation('sta2', position='10,20,20', range='5')

sta3 = net.addStation('sta3', position='10,30,10', range='5')

sta4 = net.addStation('sta4', position='10,40,20', range='5')

# Configuring WiFi nodes

net.configureWifiNodes()

# Adding links

net.addLink(ap1, ap2)

net.addLink(sta1, ap1)

net.addLink(sta2, ap1)

net.addLink(sta3, ap2)

net.addLink(sta4, ap2)

# Starting Mininet-WiFi

net.plotGraph()

net.start()

# Start Wireshark for capturing traffic on sta1

interface = 'ap1-wlan1'

start\_wireshark(interface)

# Open Mininet-WiFi CLI

CLI(net)

# Stopping Mininet-Wi

net.stop()

setLogLevel('info')

create()

 6.roam between access points(3a,6s)

import sys

from mininet.node import Controller

from mininet.log import setLogLevel, info

from mn\_wifi.net import Mininet\_wifi

from mn\_wifi.cli import CLI

def myNetwork():

net = Mininet\_wifi()

info('\* Adding controller\n')

net.addController('c0')

info('\* Add switches/APs\n')

ap1 = net.addAccessPoint('ap1', ssid='ap1-ssid', channel='1', mode='g', position='377.0,100.0,0')

ap2 = net.addAccessPoint('ap2', ssid='ap2-ssid', channel='1', mode='g', position='453.0,101.0,0')

ap3 = net.addAccessPoint('ap3', ssid='ap3-ssid', channel='1', mode='g', position='527.0,102.0,0')

info('\* Add hosts/stations\n')

sta1 = net.addStation('sta1', position='217.0,242.0,0')

sta2 = net.addStation('sta2', position='303.0,241.0,0')

sta3 = net.addStation('sta3', position='387.0,246.0,0')

sta4 = net.addStation('sta4', position='462.0,247.0,0')

sta5 = net.addStation('sta5', position='568.0,235.0,0')

sta6 = net.addStation('sta6', position='643.0,235.0,0')

info("\* Configuring Propagation Model\n")

net.setPropagationModel(model="logDistance", exp=3)

info("\* Configuring wifi nodes\n")

net.configureWifiNodes()

info('\* Add links\n')

net.addLink(ap1, ap2)

net.addLink(ap2, ap3)

net.plotGraph(max\_x=1000, max\_y=1000)

net.pingAll()

net.startMobility(time=0)

net.mobility(sta1, 'start', time=1, position='217.0,242.0,0')

net.mobility(sta2, 'start', time=2, position='303.0,241.0,0')

net.mobility(sta3, 'start', time=3, position='387.0,246.0,0')

net.mobility(sta4, 'start', time=4, position='462.0,247.0,0')

net.mobility(sta5, 'start', time=5, position='568.0,235.0,0')

net.mobility(sta6, 'start', time=6, position='643.0,235.0,0')

net.mobility(sta1, 'stop', time=11, position='655.0,235.0,0')

net.mobility(sta2, 'stop', time=12, position='655.0,235.0,0')

net.mobility(sta3, 'stop', time=13, position='655.0,235.0,0')

net.mobility(sta4, 'stop', time=14, position='655.0,235.0,0')

net.mobility(sta5, 'stop', time=15, position='255.0,242.0,0')

net.mobility(sta6, 'stop', time=16, position='217.0,242.0,0')

net.stopMobility(time=18)

net.start()

info('\* Post configure nodes\n')

CLI(net)

net.stop()

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

setLogLevel('info')

myNetwork()

1.Custom Topology Creation in Mininet(3s,6h)

from mininet.net import Mininet

from mininet.node import Controller, RemoteController, OVSController

from mininet.node import CPULimitedHost, Host, Node

from mininet.node import OVSKernelSwitch, UserSwitch

from mininet.node import IVSSwitch

from mininet.cli import CLI

from mininet.log import setLogLevel, info

from mininet.link import TCLink, Intf

from subprocess import call

def myNetwork():

net = Mininet( topo=None,

build=False,

ipBase='10.0.0.0/8')

info( '\*\*\* Adding controller\n' )

info( '\*\*\* Add switches\n')

s1 = net.addSwitch('s1', cls=OVSKernelSwitch, failMode='standalone')

s2 = net.addSwitch('s2', cls=OVSKernelSwitch, failMode='standalone')

s3 = net.addSwitch('s3', cls=OVSKernelSwitch, failMode='standalone')

info( '\*\*\* Add hosts\n')

h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)

h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)

h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)

h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)

h5 = net.addHost('h5', cls=Host, ip='10.0.0.5', defaultRoute=None)

h6 = net.addHost('h6', cls=Host, ip='10.0.0.6', defaultRoute=None)

info( '\*\*\* Add links\n')

net.addLink(s1, h2)

net.addLink(s1, h1)

net.addLink(s1, s2)

net.addLink(s2, s3)

net.addLink(s3, h6)

net.addLink(s3, h5)

net.addLink(s2, h4)

net.addLink(s2, h3)

info( '\*\*\* Starting network\n')

net.build()

info( '\*\*\* Starting controllers\n')

for controller in net.controllers:

controller.start()

info( '\*\*\* Starting switches\n')

net.get('s1').start([])

net.get('s2').start([])

net.get('s3').start([])

info( '\*\*\* Post configure switches and hosts\n')

CLI(net)

if \_\_name\_\_ == "\_\_main\_\_":

setLogLevel('info')

myNetwork()

2.Mininet-WiFi with multiple access points and mobile stations

from mininet.log import setLogLevel, info

from mininet.node import OVSSwitch, Controller

from mininet.cli import CLI

from mn\_wifi.net import Mininet\_wifi

def create():

net=Mininet\_wifi(switch=OVSSwitch ,controller=Controller)

net.addController('c0')

# acess points

ap1=net.addAccessPoint('ap1', mode='g',position='20,20,10',range='20')

ap2=net.addAccessPoint('ap2', mode='g',position='40,20,20',range='20')

#stations

sta1=net.addStation('sta1',position='10,10,10',range='5')

sta2=net.addStation('sta2',position='10,20,20',range='5')

sta3=net.addStation('sta3',position='10,30,10',range='5')

sta4=net.addStation('sta4',position='10,40,20',range='5')

#configuring

net.configureWifiNodes()

#ading links

net.addLink(ap1,ap2)

net.addLink(sta1,ap1)

net.addLink(sta2,ap1)

net.addLink(sta3,ap2)

net.addLink(sta4,ap2)

#starting

net.plotGraph()

net.start()

CLI(net)

net.stop()

setLogLevel('info')

create()

3.Simple Web Server and Client

from mininet.net import Mininet

from mininet.node import OVSController

from mininet.log import setLogLevel, info

from mininet.cli import CLI

from mininet.link import TCLink

def run():

# Create the network

net = Mininet(controller=OVSController, link=TCLink)

# Add hosts and switch

h1 = net.addHost('h1', ip='10.0.0.1/24')

h2 = net.addHost('h2', ip='10.0.0.2/24')

s1 = net.addSwitch('s1')

c0 = net.addController('c0')

# Add links

net.addLink(h1, s1)

net.addLink(h2, s1)

# Start the network

net.start()

# Create the HTML content

h1.cmd('mkdir -p /var/www/html')

h1.cmd('echo "Hello, Lubna!" > /var/www/html/index.html')

# Start the web server on h1

h1.cmd('python3 -m http.server 80 --directory /var/www/html &')

# Ensure curl is installed on h2

h2.cmd('apt-get install -y curl')

# Test the connection from h2

result = h2.cmd('curl 10.0.0.1')

info(result)

# Start the CLI for further testing

CLI(net)

# Stop the network

net.stop()

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

setLogLevel('info')

run()

4.multiple access points and mobile stations. Emulate a realistic environment where stations move between different access points

from mininet.node import Controller, OVSSwitch

from mininet.log import setLogLevel, info

from mn\_wifi.net import Mininet\_wifi

from mn\_wifi.cli import CLI

from time import sleep

def myNetwork():

net = Mininet\_wifi(switch=OVSSwitch, controller=Controller)

info('\* Adding controller\n')

net.addController('c0')

info('\* Add switches/APs\n')

ap1 = net.addAccessPoint('ap1', mode='g', position='394.0,117.0,0', range='300')

ap2 = net.addAccessPoint('ap2', mode='g', position='546.0,119.0,0', range='300')

info('\* Add hosts/stations\n')

sta1 = net.addStation('sta1', position='267.0,265.0,0', range='300')

sta2 = net.addStation('sta2', position='371.0,268.0,0', range='300')

sta3 = net.addStation('sta3', position='510.0,293.0,0', range='300')

sta4 = net.addStation('sta4', position='629.0,276.0,0', range='300')

info("\* Configuring wifi nodes\n")

net.configureWifiNodes()

info('\* Add links\n')

net.addLink(ap2, ap1)

net.plotGraph(max\_x=1000, max\_y=1000)

net.startMobility(time=0)

net.mobility(sta1, 'start', time=1, position='267.0,265.0,0')

net.mobility(sta2, 'start', time=2, position='371.0,268.0,0')

net.mobility(sta1, 'stop', time=10, position='510.0,293.0,0')

net.mobility(sta2, 'stop', time=10, position='510.0,293.0,0')

net.mobility(sta3, 'start', time=3, position='510.0,293.0,0')

net.mobility(sta3, 'stop', time=10, position='267.0,265.0,0')

net.stopMobility(time=11)

net.start()

net.pingAll()

CLI(net)

net.stop()

setLogLevel('info')

myNetwork()